

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення комп'ютерних та інформаційно-пошукових систем»**

спеціальності 121 Інженерія програмного забезпечення

**на тему: «Постачальник контенту для навчальних платформ Learnsmart
та Smartbook»**

Виконав:

студент IV курсу, групи КП-62

Благодатний Дмитро Романович _____

Керівник:

Старший викладач кафедри ПЗКС, к.т.н.,

Бухтіяров Юрій Вікторович _____

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович _____

Рецензент:

Доцент кафедри ЕІ, к.т.н., доцент,

Вунтесмері Юрій Володимирович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення комп'ютерних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2019 р.

ЗАВДАННЯ

на дипломний проєкт студенту

Благодатному Дмитру Романовичу

1. Тема проєкту «Постачальник контенту для навчальних платформ Learnsmart та Smartbook», керівник проєкту Бухтіяров Юрій Вікторович, ст. викладач, к.т.н., затверджені наказом по університету від «25» травня 2020 р. №1181-с
2. Термін подання студентом проєкту «17» червня 2020 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - збір та опис вимог до вебзастосунку для планування та трекінгу задач;
 - аналіз мов програмування та технологій розроблення веб-сервісів;
 - розроблення вебзастосунку для планування та трекінгу задач;
 - аналіз розробленого вебзастосунку для планування та трекінгу задач.
5. Перелік обов'язкового графічного матеріалу:
 - діаграма використання системи (креслення);
 - схема бази даних (креслення);
 - інтерфейс користувача (плакат);

– архітектура системи (плакат).

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

7. Дата видачі завдання «31» жовтня 2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	10.11.2019	
2.	Розроблення та узгодження технічного завдання	27.11.2019	
3.	Розроблення структури вебзастосунку	14.12.2019	
4.	Підготовка матеріалів першого розділу дипломного проєкту	27.12.2019	
5.	Розроблення дизайну сторінок та графічних елементів	02.02.2020	
6.	Підготовка матеріалів другого розділу дипломного проєкту	19.02.2020	
7.	Програмна реалізація вебзастосунку	12.03.2020	
8.	Тестування вебзастосунку	20.03.2020	
9.	Підготовка матеріалів третього розділу дипломного проєкту	04.04.2020	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	24.04.2020	
11.	Підготовка графічної частини дипломного проєкту	12.05.2020	
12.	Оформлення документації дипломного проєкту	24.05.2020	

Студент

Дмитро БЛАГОДАТНИЙ

Керівник проєкту

Юрій БУХТІЯРОВ

АНОТАЦІЯ

Дипломний проєкт присвячено розробці програмного забезпечення для постачання контенту для навчальних платформ.

В даному проєкті було розроблено архітектуру серверної та клієнтської частини вебзастосунку, структуру бази даних та створено дизайн інтерфейсів програмного забезпечення.

Вебзастосунок, що було розроблено, надає користувачам можливість конвертувати будь-який PDF-файл у формат “.bnn”, з яким вже безпосередньо може працювати система менеджменту навчання. Дана система управління навчальною діяльністю використовується для розробки, управління та поширення навчальних онлайн-матеріалів із забезпеченням спільного доступу. Створюються навчальні ресурси у візуальному навчальному середовищі з завданням послідовності вивчення.

До складу системи входять засновані як на змістовній так і на комунікативній компоненті індивідуальні завдання, проекти для роботи в малих групах, а також навчальні елементи для всіх студентів.

ABSTRACT

The diploma project is devoted to the development of software for the supply of content for educational platforms.

In this project, the architecture of the server and client part of the web application, the structure of the database and the design of software interfaces were developed.

The developed web application allows users to convert any PDF file to ".bnn" format, with which the learning management system can already work directly. This learning management system is used to develop, manage and disseminate online learning materials. Learning resources are created in the visual learning environment with the task of sequence of study.

The system includes individual tasks based on both the content and the communicative component, projects for work in small groups, as well as learning elements for all students.

АННОТАЦИЯ

Дипломный проект посвящен разработке программного обеспечения для поставки контента для учебных платформ.

В данном Проекте была разработана архитектура серверной и клиентской части веб-приложения, структура базы данных и создан дизайн интерфейсов программного обеспечения.

Веб-приложение, что было разработано, предоставляет пользователям возможность конвертировать любой файл PDF в формат ".bnp", с которым уже непосредственно может работать система менеджмента обучения. Данная система управления учебной деятельностью используется для разработки, управления и распространения учебных онлайн-материалов с обеспечением совместного доступа. Создаются учебные ресурсы в визуальной учебной среде с заданием последовательности изучения.

В состав системы входят основанные как на содержательной так и на коммуникативной компоненте индивидуальные задания, проекты для работы в малых группах, а также учебные элементы для всех студентов.

ДП.045440-01-90 Постачальник контенту для навчальних контенту. Відомість проєкту

Позначення	Найменування	Кіл-ть	Примітка
	Документація проєкту		
ДП.045440-02-91	Постачальник контенту	5	
	для навчальних		
	платформ. Технічне		
	завдання		
ДП.045440-03-81	Постачальник контенту	50	
	для навчальних		
	платформ. Пояснювальна		
	записка		
ДП.045440-04-51	Постачальник контенту	4	
	для навчальних		
	платформ. Програма та		
	методика тестування		
ДП.045440-05-34	Постачальник контенту	10	
	для навчальних		
	платформ. Керівництво		
	користувача		
ДП.045440-06-99	Постачальник контенту	1	
	для навчальних		
	платформ. Діаграма		
	використання. Схема		
	алгоритму		
ДП.045440-07-99	Постачальник контенту	1	
	для навчальних		
	платформ. База даних		
	системи. Схема даних		

[illegible]

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2019 р.

ПОСТАЧАЛЬНИК КОНТЕНТУ ДЛЯ НАВЧАЛЬНИХ ПЛАТФОРМ
LEARNSMART ТА SMARTBOOK

Технічне завдання

ДП.045440-02-91

«ПОГОДЖЕНО»

Керівник проєкту:

_____ Юрій БУХТІЯРОВ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Дмитро БЛАГОДАТНИЙ

ЗМІСТ

1. Найменування та галузь застосування	3
2. Підстава для розроблення.....	3
3. Призначення розробки	3
4. Вимоги до програмного продукту.....	3
5. Вимоги до проєктної документації	4
6. Етапи проєктування	5
7. Порядок тестування розробки	5

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Постачальник контенту для навчальних платформ Learnmsrt та Smartbook.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проєктування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання в якості інформаційне забезпечення для об'єднання та покращення процесу обліку волонтерських робіт учнів навчальних закладів.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Платформа повинна забезпечувати такі основні функції:

- 1) можливість динамічного оновлення вмісту вебсторінок;
- 2) авторизований доступ до платформи;
- 3) можливість створювати, редагувати та видаляти ресурси;
- 4) можливість присвоювати проби до маркованого тексту;
- 5) підтримка вебзастосунку та десктоп-застосунку;
- 6) дозволяє авторам навчальних ресурсів розміщувати матеріал на навчальних платформах Learnmsmart та Smartbook.

Розробку виконати на платформі .NET з використанням технології AJAX.

Додаткові вимоги:

- 1) наявність динамічного меню;

- 2) наявність анімованих кнопок;
- 3) адаптивний дизайн для вебзастосунку;
- 4) дизайн сторінок з використанням в якості базових білого та зеленого кольорів

5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проєкту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Компоненти платформи»;
 - «Структура бази даних».

6. ЕТАПИ ПРОЄКТУВАННЯ

Вивчення літератури за тематикою роботи.....	14.11.2019
Розроблення та узгодження технічного завдання.....	28.11.2019
Розроблення структури платформи.....	15.12.2019
Розроблення дизайну сторінок та графічних елементів.....	03.02.2020
Програмна реалізація платформи.....	17.03.2020
Тестування платформи.....	03.04.2020
Підготовка матеріалів текстової частини проєкту.....	28.04.2020
Підготовка матеріалів графічної частини проєкту.....	12.05.2020
Оформлення технічної документації проєкту.....	25.05.2020

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«__» _____ 2020 р.

ПОСТАЧАЛЬНИК КОНТЕНТУ ДЛЯ НАВЧАЛЬНИХ ПЛАТФОРМ
LEARNSMART ТА SMARTBOOK

Пояснювальна записка

ДП.045440-03-81

«ПОГОДЖЕНО»

Керівник проєкту:

_____ Юрій БУХТІЯРОВ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Дмитро БЛАГОДАТНИЙ

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ	3
ВСТУП	6
1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	8
1.1. Загальний опис проблеми	8
1.2. Аналіз існуючих рішень для даної задачі	9
1.3. Постановка задачі	11
1.4. Висновки.....	12
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .	13
2.1. Обґрунтування вибору типу програмного забезпечення	13
2.1. Аналіз та порівняння RDBMS.....	16
2.2. Обґрунтування вибору мови програмування для розробки серверної частини застосунку	20
2.3. Обґрунтування вибору мови програмування для розробки клієнтської частини застосунку	23
3. ОПИС СТРУКТУРО-АЛГОРИТМІЧНОЇ ОРГАНІЗАЦІЇ ВЕБЗАСТОСУНКУ.....	30
3.1. Загальний опис системи.....	30
3.2. Опис вимог до функціональності вебзастосунку	31
3.3. Архітектура вебзастосунку.....	32
3.4. Опис модулів ПЗ.....	36
3.5. Висновки.....	36
4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ	38
4.1. Робочий потік вебзастосунку	38
4.2. Тестування вебзастосунку	42
4.3. Порівняння розробки із існуючими аналогами	43
4.4. Рекомендації щодо подальшого вдосконалення	43
4.5. Висновки.....	44
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ.....	50

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

AI – Artificial Intelligence (розділ комп'ютерної лінгвістики та інформатики, що опікується формалізацією проблем та завдань, які подібні до дій, що виконує людина).

ПК – Персональний Комп'ютер.

LMS – Learning Management System (система управління навчальною діяльністю, яка використовується для розробки, управління та поширення навчальних онлайн-матеріалів із забезпеченням спільного доступу).

e-Learning – описує когнітивні принципи ефективного мультимедійного навчання з використанням електронних освітніх технологій.

GPL – General Public License.

SCORM – Shareable Content Object Reference Model (набір стандартів та специфікацій, розроблений для систем дистанційного навчання).

PDF – Portable Document Format.

HTML – Hypertext Markup Language.

OAuth – Open Authorization.

RDBMS – Relational Database Management System.

JDBC – Java Database Connectivity.

БД – База Даних.

ПЗ – Програмне Забезпечення.

SQL – Structured Query Language.

СУБД – Система Управління Базами Даних.

API – Application Program Interface.

CPU – Central Processing Unit.

ОС – Операційна Система.

4G – стандарт четвертого покоління мобільного радіозв'язку.

HTTPS – Hypertext Transfer Protocol Secure (захищений протокол передачі даних).

SSL – Secure Sockets

Layer (криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером).

USB – Universal Serial Bus (призначена для з'єднання комп'ютерів і периферійних пристроїв).

.NET – програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і вебзастосунків.

DOM – Document Object Model (специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами).

JS – JavaScript.

XAML – Extensible Application Markup Language (спрощує створення користувацького інтерфейсу для програми .NET Framework).

XML – Extensible Markup Language (запропонований консорціумом World Wide Web Consortium (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками).

API – Application Program Interface.

LO – Learning Objective.

dpi – dots per inch (показник роздільної здатності при введенні чи виведенні зображальної інформації на/в носій для подальшої її виведення, обробки чи зберігання).

EPUB – electronic publication (відкритий стандарт формату електронних книг від International Digital Publishing Forum).

Проба – питання, призначене для перевірки знань та навичок учня, визначених у навчальній меті (LO). Проба повинна відповідати навчальній меті, як зі сторони пізнавальності, так і зі сторони необхідності.

Heat Map – одна із основних функцій LMS, яка допомагає зрозуміти актуальність використання та розуміння студентами як окремих LO, так і всього розділу цілком.

MVC – Model-view-controller (передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та

модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача).

ВСТУП

Адаптивне навчання – це навчальний метод, який використовує комп'ютерні алгоритми, щоб організувати взаємодію з учнем, та забезпечити спеціальні ресурси та навчальну діяльність для задоволення унікальних потреб кожного учня. Дані алгоритми адаптують навчальний матеріал відповідно до потреб учнів, на що вказують їх відповіді на запитання, завдання та досвід. Ця технологія охоплює різні галузі, в тому числі інформатику, AI, психометрію, освіту, психологію.

Адаптивне навчання виникло частково із-за усвідомлення того, що індивідуальне навчання неможливо впровадити широкомасштабно, використовуючи традиційні підходи. Адаптивні системи навчання намагаються перетворити учня з пасивного рецептора інформації на співучасника в навчальному процесі. Освіта – основне застосування адаптивних систем навчання. Вони були розроблені як застосунки для домашніх ПК, веббраузерів, і тепер їх впроваджують у загальні навчальні платформи.

Інтелектуальне (адаптивне) навчання започаткувалось від AI, та почало набирати популярність в 1970-х. У той час було прийнято вважати, що комп'ютери зрештою досягнуть високого рівня здатності людини до адаптивності. Тим не менш, основним бар'єром для впровадження адаптивного навчання були розміри комп'ютерів та витрати на них, що робило широке застосування непрактичним. Ще однією перешкодою було те, що користувацькі інтерфейси не сприяли навчальному процесу.

Метою цієї дипломної роботи є розширити дослідження в галузі адаптивного навчання шляхом аналізу існуючих застосунків та комп'ютерних алгоритмів, що лежать в основі інтелектуального навчання, а також вдосконалення методів для зменшення затрат ресурсів на розв'язок головної задачі.

У цьому дипломному проєкті планується проаналізувати існуючі реалізації методів адаптивного навчання та їх модифікації з визначенням їх ефективності.

1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1. Загальний опис проблеми

LMS – це одна із імплементацій принципів адаптивного навчання. Її концепція виникла безпосередньо з e-Learning. LMS була розроблена для виявлення прогалин у процесі навчання, використовуючи аналітичні дані та звітність. Сучасні LMS включають в собі адаптивні алгоритми для автоматизованих рекомендацій в навчальних курсах на основі профілю навичок користувача, а також мають змогу діставати метадані із навчальних матеріалів, щоб зробити такі рекомендації ще більш точними.

LMS постачає та керує всіма видами навчального контенту, включаючи відео, курси та документи. За допомогою LMS викладачі можуть:

- створювати та інтегрувати матеріали навчального курсу;
- налаштовувати його зміст та оцінки за певні навчальні об'єкти курсу (лекції, практики і т.д.);
- відстежувати прогрес навчання;
- створювати тест індивідуально для кожного студента.

LMS дозволяє налагодити взаємокомунікацію між різними навчальними об'єктами та організовувати графіки навчання. Такі системи мають вбудовані функції, включаючи систему оцінювання та алгоритм відстеження прогресу студента. Таким чином, учні можуть бачити в реальному часі їх прогрес, а викладачі можуть контролювати та повідомляти про ефективність навчання.

Однією з найважливіших особливостей LMS є створення спрощеної комунікації між учнями та викладачами. Такі системи, окрім сприяння навчанню в інтернеті, відстеження прогресу навчання, надання цифрових засобів навчання, також можуть використовуватися для надання різних комунікаційних функцій.

Головна проблема – складність використання LMS застосунків для початківців. Різні компанії пропонують свої реалізації того чи іншого алгоритму, немає загальних рекомендацій які б регулювали це питання. Перша причина – розробка такого програмного забезпечення потребує добре сформованої та обгрунтованої технологічної структури. Також створення такого програмного продукту навіть в початковому вигляді потребує достатньо багато часу. Другою причиною є наявність бажання у викладача трансформувати свої лекції із тет-а-тет формату в онлайн. Часто навіть досвідчений вчитель не може одразу відмовитись від своїх традиційних методів викладання, які вже сформувались з плином часу.

1.2. Аналіз існуючих рішень для даної задачі

Проаналізувавши готові LMS застосунки, які реалізують алгоритми адаптивного навчання, було помічено, що кожне з цих програмних забезпечень має недоліки. Основним серед них є те, що для роботи із відповідним ПЗ необхідно витратити чимало часу, а в деяких випадках не обійтись навіть без спеціальних технічних знань. Нижче наведено найбільш поширені сервіси LMS, а також їх переваги та недоліки.

1.2.1. Chamilo

Chamilo – це безкоштовне програмне забезпечення (має GPL ліцензію) для онлайн навчання та управління навчальним контентом, спрямоване на покращення доступу до освіти та знань у всьому світі. Підтримується асоціацією Chamilo, яка має за ціль, крім просування власного програмного забезпечення, ще і підтримку чіткого каналу зв'язку та побудову глобальної мережі постачальників своїх послуг і розробників програмного забезпечення.

Має наступні переваги:

- структуровані курси та навчальні цикли;
- власна соціальна мережа для користувачів з інтеграцією навчальних об'єктів;
- підтримка стандарту SCORM 1.2;

- мультиуніверситетський режим (з централізованим управлінням);
- перевірена підтримка більш ніж 700000 користувачів.

Також має наступні недоліки:

- обов'язкова наявність застарілого Flash-плагіна у веббраузері;
- тісна взаємодія з third-party сервісами, із-за чого дані можуть бути незахищеними;
- відсутність можливості зберегти свої навчальні об'єкти на хмарне сховище;
- остання стабільна версія випускалась 8-ого травня 2019 року.

1.2.2. Claroline

Claroline організована навколо концепції вільного простору в створенні курсів та педагогічної діяльності. Кожен простір курсу містить перелік інструментів, який дозволяє викладачу:

- публікувати навчальні ресурси в будь-якому форматі (текст, PDF, HTML, відео і т.д.);
- структурувати порядок денний із завданнями та термінами;
- відсилати сповіщення (в тому числі і електронною поштою);
- використовувати вікіпедію задля написання спільних документів.

Тим не менш, Claroline має наступні недоліки:

- останній стабільний випуск продукт відбувся аж 28-ого лютого 2006 року;
- викладач, що працює в декількох навчальних закладах, повинен створювати окремий акаунт для кожного з них;
- відповідно до попереднього пункту, викладач не має змогу напряму поділитися власними матеріалами з іншими своїми акаунтами, а має вручну копіювати їх.

1.2.3. eFront

eFront призначений для сприяння створенню онлайн-навчальних спільнот, пропонуючи різні можливості для співпраці та взаємодії через користувацький інтерфейс на основі іконок. Платформа пропонує інструменти для створення контенту, складання тестів, управління завданнями, а також звітування, внутрішні повідомлення, форум, чат, опитування, календар та інше. Має наступні переваги:

- конструктор екзаменів та завдань;
- інтеграція платежів;
- підтримка стандарту SCORM 1.2;
- мультиуніверситетський режим (з централізованим управлінням).

Недоліки:

- не дивлячись на інтеграцію платежів, сервіс не дозволяє авторизуватись через OAuth;
- відсутність можливості зберегти свої навчальні об'єкти на хмарне сховище.

1.2.4. Результати аналізу

Програмного застосунку, який би не мав суттєвих недоліків, які наведені у пунктах вище, не існує. Виходячи з отриманих даних після проведення аналізу, всі застосунки є різними і мають різну складність і швидкодію. Кожен із них по-різному підходить для вирішення задачі. Тож основною задачею є програмна реалізація вищенаведених пунктів і створення алгоритму, що буде якнайкраще підходити до сучасних реалій і можливостей адаптивного навчання.

1.3. Постановка задачі

В результаті проведеного дослідження можна сформулювати такі вимоги до дипломної роботи:

- реалізація загального алгоритму адаптивного навчання;

- розроблення програмного застосунку;
- реалізація графічно-користувачького інтерфейсу;
- дослідження існуючих реалізацій методів адаптивного навчання аналітичним шляхом;
- загальне тестування системи на правильність даних;
- тестування алгоритмів інтелектуального навчання на різних наборах даних;

Головний результат роботи – успішна реалізація загального алгоритму адаптивного навчання для більшої відповідності функціональності програмного застосунку відповідно до сучасних вимог.

1.4. Висновки

У першій частині цього розділу наведено загальний опис роботи LMS та їх призначення. Проаналізовано принцип роботи даного програмного застосунку. Також обґрунтовано проблеми, пов'язані з використанням LMS.

У другій частині поточного розділу наведено існуючі реалізації методів адаптивного навчання: Chamilo, Claroline, eFront, а також обґрунтовано їх складність та що кожен із цих програмних застосунків по-різному підходить до вирішення головної задачі.

У третій частині поточного розділу сформовано вимоги до даної дипломної роботи. Вимоги стосуються програмної та логічної частин.

2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Вибір засобів реалізації є одним із основних кроків при розробленні програмного забезпечення. Для даного застосунку необхідно обрати RDBMS, мову програмування та фреймворк для розроблення, а також визначитись із типом застосунку та технологіями клієнтської частини.

2.1. Обґрунтування вибору типу програмного забезпечення

2.1.1. Вебзастосунок

Вебзастосунок – розподілений застосунок, в якому клієнтом виступає браузер, а сервером – вебсервер. Доступ до вебзастосунків користувач здійснює через веббраузер із активним підключенням до інтернету.

Посилення використання інтернет-технологій серед корпорацій та приватних підприємців вплинуло на спосіб ведення бізнесу. Це призвело до широкого впровадження та наступного застосування вебзастосунків, оскільки компанії поступово відмовляються від традиційних моделей ведення бізнесу, і надають перевагу вебтехнологіям. Вебзастосунки дають підприємствам можливість впорядкувати свої операції, підвищити ефективність та зменшити витрати.

Вебзастосунки, такі як клієнти електронної пошти, текстові процесори, електронні таблиці та інші, забезпечують таку ж функціональність, як і десктоп-версії для ПК. А також мають наступні переваги:

- вебзастосунки працюють на багатьох платформах незалежно від ОС або пристрою;
- усі користувачі будь-якого вебзастосунку отримують доступ до єдиної, актуальної версії, усуваючи проблему сумісності;

- вебпрограми не встановлюються на жорсткому диску ПК, тож не залежать від об'єму комп'ютерної пам'яті;
- зменшують витрати як для бізнесу, так і для кінцевого користувача, оскільки бізнес більше не потребує широкої підтримки та обслуговування продукту, а користувач – високі вимоги до ПЗ;
- просте налаштування. Потрібно лиш пару хвилин щоб налаштувати нового користувача та надати йому доступ.

Тим не менш, вебзастосунки мають наступні недоліки:

- підтримка різних браузерів. Під час розробки вебзастосунку потрібно враховувати, що кінцевий користувач може запустити його у іншому браузері. Тож варто надійно тестувати вебзастосунок на різних веббраузерах, серед яких основними є Google Chrome, Mozilla Firefox, Safari та Microsoft Edge;
- продуктивність застосунку. Порівняно з нативними десктоп-версіями одного і того ж застосунку, вебзастосунок працює повільніше, оскільки десктоп-застосунок передбачає локальний запуск програми, що забезпечує більш швидкий доступ до необхідних для програми ресурсів: текстур, скриптів і т.д.;
- підтримка швидкості інтернету. Не дивлячись на те, що доступ до Wi-Fi та 4G наявний у переважній більшості місць земної кулі, втрачений зв'язок з інтернетом завадить роботі з вебзастосунком;
- безпека. Для надійної та безперебійної роботи вебзастосунку з даними користувача сайту потрібно використовувати відповідні протоколи передачі даних, такі як HTTPS або SSL. Але вони потребують додаткової конфігурації.

2.1.2. Нативний десктоп-застосунок

Десктоп-застосунок – користувацька комп'ютерна програма, що дає змогу вирішувати конкретні прикладні задачі користувача.

В залежності від виду діяльності, для якої був розроблений нативний застосунок (прикладна програма, прикладне ПЗ), він може маніпулювати текстом, звуком, графікою, і, як результат, комбінацією цих елементів – це забезпечує повну інтеграцію виду діяльності у відповідну функціональність десктоп-застосунку.

Деякі нативні програми фокусуються на одному завданні, наприклад, обробці тексту. Інші, що називаються інтегрованим програмним забезпеченням, включають в собі декілька програм.

Нативні десктоп-застосунки мають наступні переваги:

- працюють в офлайн режимі. Це допомагає користувачеві безперебійно взаємодіяти з прикладною програмою в місцях, де інтернет-з'єднання відсутнє або має дуже низьку швидкість (літаки і т.д.);
- налагоджений доступ до системних ресурсів. Нативні десктоп-застосунки набагато швидше взаємодіють з апаратними ресурсами ПК, такими як USB, графічним процесором, приймачами Bluetooth та іншими;
- розширений доступ до функцій ОС. Прикладні програми можуть використовувати такі ресурси ОС, до яких вебзастосункам важко або неможливо отримати доступ (наприклад, система сповіщень).

Крім того, десктоп-застосунки мають наступні недоліки:

- встановлення та подальше користування нативним додатком. Для встановлення прикладної програми потрібно виконати багато кроків: відкрити вебсторінку застосунку, знайти посилання для скачування, прийняти всі положення та умови виробника і т.д. Більша частина користувачів не докладає таких зусиль, і в результаті віддають перевагу вебзастосункам;

- десктоп-застосунок на різних ОС. В залежності від платформи, для якої потрібно розробити нативний застосунок, потрібно наймати окрему відповідальну команду;
- час розробки. Відштовхуючись від попереднього пункту, можна дійти висновку, що у разі розроблення прикладної програми для різних ОС потрібно фактично створити окремі програми для кожної із них, що займає в рази більше часу, ніж при розробці вебзастосунку;
- нативні застосунки потребують частого оновлення.

2.1.3. Результати аналізу

Врахувавши переваги та недоліки кожного із типів програмного забезпечення, було вирішено, що в цій роботі буде доцільно використовувати вебзастосунок як вид ПЗ, оскільки саме цей тип більш гнучкий у виборі технологій розробки, має менші витрати на розробку програмного коду, а також вебзастосунки легше оновлювати при виявленні помилок (у випадку з нативним додатком потрібно надсилати оновлений код у магазин застосунків та сподіватися, що користувачі встановлять оновлення).

2.1. Аналіз та порівняння RDBMS

Реляційна база даних – база даних, заснована на реляційній моделі даних. Ця модель впорядковує дані в одну або кілька таблиць стовпців і рядків, з унікальним ключем, що ідентифікує кожен рядок. Рядки також називають записами або кортежами, стовпці – атрибутами. Як правило, кожна таблиця представляє один “тип сутності” (наприклад, “замовник” або “товар”). Рядки представляють екземпляри цього типу сутності (наприклад, “Лі” або “стілець”) та стовпці, що представляють значення, приписані цьому екземпляру (наприклад, “адреса” або “ціна”).

З 1980-х років RDBMS були поширеним варіантом для зберігання інформації в базах даних, що використовуються для фінансових записів,

виробничої та матеріально-технічної інформації, персональних даних і т.д. Реляційні бази даних часто замінювали застарілі ієрархічні бази даних та мережеві бази даних, оскільки RDBMS було легше впроваджувати та адмініструвати.

2.2.1. MySQL

MySQL – це реляційна система керування базами даних, яка знаходиться у вільному доступі. MySQL належить та спонсорується шведською компанією MySQL AB, яку придбала компанія Sun Microsystems. MySQL використовується у багатьох вебдодатках, таких як Drupal, Joomla, phpBB та WordPress.

MySQL можна будувати та встановлювати вручну з вихідного коду, але він частіше встановлюється з бінарного пакету, якщо не потрібні спеціальні налаштування. У більшості дистрибутивів Linux система управління пакетами може завантажувати та встановлювати MySQL з мінімальними зусиллями, хоча для налаштування параметрів безпеки та оптимізації часто потрібна додаткова конфігурація.

Багато мов програмування включають в собі бібліотеки для доступу до баз даних MySQL. Сюди входить MySQL Connector для інтеграції з Visual Studio Microsoft (найчастіше використовуються такі мови, як C # і VB) та драйвер JDBC для Java.

MySQL – це стабільне, надійне та потужне ПЗ з багатьма перевагами, такими як:

- безпека даних;
- масштабованість при потребі розробника;
- висока продуктивність;
- кешування запитів;
- підтримка повнотекстового пошуку та індексування;
- групування коммітів;
- крос-платформенна підтримка;
- вбудована бібліотека баз даних;

- вкладені SQL-запити;
- вбудована підтримка реплікації.

Втім, серед головних недоліків MySQL можна вказати наступне:

- ранні версії MySQL (5.0 і раніше) не підтримують операції `ROLE`, `COMMIT` та процедури, що можна зберігати на самому сервері MySQL;
- недостатньо ефективно підтримує великі розміри БД;
- не має якісного середовища для розробки, а також зрозумілих інструментів для дебагу;
- схильний до неефективної обробки транзакцій, що призводить до пошкодження даних.

2.2.2. Microsoft SQL Server

Microsoft SQL Server – це реляційна СУБД, яка розробляється та підтримується компанією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі.

Усі операції, які можна викликати на SQL Server, передаються йому через визначений компанією Microsoft формат, який називається Tabular Data Stream (TDS) – табличний потік даних. TDS – протокол рівня застосунків, який використовується для передачі даних між сервером бази даних та клієнтом. Але і не зважаючи на цей стандартизований протокол, зі Microsoft SQL Server можна взаємодіяти через API SQL Server, що є достатньо зручним для вебзастосунків.

SQL Server буферизує дані в оперативній пам'яті для мінімізації операції вводу-виводу диска. Будь-яка сторінка розміром в 8 Кб може бути буферизована в пам'яті, а її обсяг, доступний для SQL Server, визначає, скільки сторінок буде кешовано. Будь-яке читання з будь-якої сторінки, чи запис на неї, копіює цю сторінку в кеш-пам'ять. Подальші читання або записи переспрямовуються на копію пам'яті, а не на версію диска.

Microsoft SQL Server має наступні, суттєві переваги:

- налагоджене та професійне ПЗ (SQL Server Enterprise Edition) для управління БД в підприємницьких цілях;
- повна підтримка тригерів;
- відмінна підтримка відновлення даних;
- доступність функцій по відновленню цілих баз даних;
- безкоштовне використання SQL Server Enterprise Edition для розробницьких чи навчальних цілей;
- може керувати базами даних розміром до 524 петабайт та підтримує 640 логічних процесорів (CPU);
- Microsoft та інші компанії пропонують велику кількість програмних засобів розробки, які дозволяють розробляти застосунки для бізнесу з використанням баз даних Microsoft SQL Server.

Втім SQL Server має наступні недоліки:

- неможливо створити або відновити резервні копії окремих таблиць;
- висока ціна за ліцензію SQL Server Enterprise Edition для бізнесу. Для малих підприємств та фізичних осіб ціна в 7171 доллар за місяць є недоступною;
- обмежена сумісність з іншими ОС. Microsoft SQL Server призначений для роботи лише на серверах на базі Windows;
- неможливість запуску SQL Server на Unix-подібних системах задля зменшення витрат на ліцензування;
- проблема сумісності з програмами, що працюють на інших ОС. Microsoft SQL Server не передбачає доступного API для них.

2.2.3. Результати аналізу

Таким чином, проаналізувавши інформацію про RDBMS Microsoft SQL Server та MySQL, а також порівнявши їх переваги та недоліки, було

виділено пріоритети Microsoft SQL Server над MySQL. У наступному списку наведемо основні з них:

- відмінно структуроване та інтуїтивно зрозуміле ПЗ у SQL Server для управління БД. Лиш кілька конкурентів (одним із них є MySQL) змогли розробити подібне ПЗ, але SQL Server Enterprise Edition є набагато простішим у використанні та має більше функцій;
- повна підтримка тригерів у SQL Server на відміну від MySQL, який лиш недавно презентував їх. Тим не менш, вони не підтримуються повністю;
- налагоджений процес відновлення даних на відміну від MySQL, який схильний до неефективної обробки транзакцій, що призводить до пошкодження даних;
- Microsoft Visual Studio має вбудовану підтримку конфігурації Microsoft SQL Server та програмування специфічних для цієї RDBMS SQL запитів.

Врахувавши переваги та недоліки кожної RDBMS, було вирішено, що в цій роботі буде доцільно використовувати саме Microsoft SQL Server.

2.2. Обґрунтування вибору мови програмування для розробки серверної частини застосунку

Після аналізу функціональних особливостей розроблюваного вебзастосунку було зроблено висновок, що найкраще його буде побудувати на основі добре відомої клієнт-серверної архітектури, а саме: створити модуль клієнтської частини для реалізації користувацького інтерфейсу, та модуль серверної частини, що відповідатиме за взаємодію із базою даних та зовнішніми сервісами. Для реалізації серверної частини вебзастосунку було вирішено обрати мову програмування з підтримкою ООП, великою кількістю стандартних бібліотеки та з наявністю вибору фреймворку для побудови вебсерверу.

2.3.1. C#

Мова C # є кращою архітектурою для програмування серверної частини програм та автоматизації в середовищі Windows. C# – лише одна з мов .NET, але її синтаксис у стилі C набагато популярніший у програмістів, які звикли до таких мов, як Java або C ++.

Проект C# був початий в грудні 1998 і отримав кодову назву COOL (C-style Object Oriented Language). Версія 1.0 була анонсована разом з платформою .NET у червні 2000 року, тоді ж з'явилася і перша загальнодоступна бета-версія. C# 1.0 остаточно вийшов разом з Microsoft Visual Studio .NET у лютому 2002 року.

Переваги:

- відмінно інтегрується з Windows ОС. Для запуску програми C# у Windows середовищі не потрібні спеціальні конфігурації;
- застосунки, написані на C#, легко розгортаються на локальному/віддаленому сервері. Будь то вебзастосунок, сервіс Windows або нативний десктоп-застосунок;
- відносно просто знайти розробників C#. Синтаксис C# дуже схожий з синтаксисом Java, тож це ще більше спрощує пошуки;
- Microsoft Visual Studio має вбудовані та інтегровані інструменти для контролю версій коду, а також інші корисні плагіни призначені спеціально для розробки на C#;
- C# – це компільована мова, тож код може зберігатися на відкритому сервері у бінарному вигляді;
- беручи до уваги попередній пункт, можна дійти висновку, що хакер не зможе отримати доступ до вихідного коду. Щоб зловмисник міг хоч якось нашкодити роботі програми, він повинен декомпілювати бінарні файли, що є неможливим.

Недоліки:

- при внесенні навіть незначних змін в програмний код потрібно перекомпілювати всю програму і знову розгортати її на

сервері. При неповному тестуванні цих самих змін це може призвести до неочікуваних помилок;

- для запуску .NET необхідною умовою є платформа Windows;
- Microsoft перестає підтримувати старі версії .NET фреймворку після мажорних оновлень Windows ОС;
- при оновленні застосунку до більш актуальної версії .NET потрібно добре протестувати програмний код, та затвердити необхідні зміни для розгортання на сервері;
- попередній пункт передбачає достатньо вагомі витрати коштів та часу, що не завжди вигідно для підприємств.

2.3.2. Java

Java – об'єктно-орієнтована мова програмування, випущена в 1995 році компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того ж року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

«Oracle» надає розробнику компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм.

Має наступні переваги:

- автоматичне управління пам'яттю;
- набір можливостей для опрацювання виключень на помилки;
- багатий набір стандартних колекцій: список, стек, масив тощо;
- уніфікований доступ до баз даних;

- неявний спосіб перетворення стандартних типів;
- підтримка лямбд, загального типу даних та ін.

До недоліків мови можна віднести:

- авторське право, в результаті якого процес покращення мови є достатньо повільним в порівнянні з мовами відкритого доступу;
- підтримка загального типу даних (англ. generics) в Java підлягає серйозній критиці, адже реалізована за допомогою стирання типу, через що справжній тип параметра не є доступним під час виконання програми;
- відсутність реалізації беззнакових типів даних, що вдвічі зменшує граничні значення великих чисел, які часто використовуються у багатьох задачах числової обробки, що робить Java програшною мовою у порівнянні з іншими для таких цілей.

2.3.3. Результати аналізу

Враховавши переваги та недоліки кожної із мов програмування, було вирішено, що в цій роботі буде доцільно використовувати C# як мову програмування для реалізації модуля серверної частини застосунку, оскільки саме ця мова програмування найкраще підходить для розробки застосунку у середовищі Windows, а також має ряд необхідних бібліотек для швидкої та ефективної розробки функціональності вебзастосунку.

2.3. Обґрунтування вибору мови програмування для розробки клієнтської частини застосунку

Основними вимогами до технології для реалізації інтерфейсу клієнтської частини застосунку є:

- тісна інтеграція з платформою Windows ОС;
- наявність достатньої кількості бібліотек для реалізації необхідної функціональності;
- інтеграція з середовищем розробки Microsoft Visual Studio.

2.4.1. JavaScript

JavaScript – це домінуюча мова для програмування клієнтської частини вебзастосунків. Файли, які мають розширення “.js”, зазвичай називають скриптами, та імпортуються через HTML-документ для взаємодії з DOM. Усі основні веббраузери мають вбудований двигун JS, який дозволяє їм виконувати код цих самих скриптів на будь-якому пристрої користувача.

В основному JS використовується для створення сценаріїв вебсторінок, а також для надання їм інтерактивності. Сучасні фреймворки, такі як React, AngularJS, Vue.js та інші, дозволяють створювати односторінкові вебзастосунки. Це надає більшої гнучкості вебсторінкам, та дозволяє користувачу відсилати запити до серверу на необхідні ресурси без повторної повної загрузки HTML-сторінки.

Також JS може використовуватись як єдина мова програмування для розробки серверних застосунків (фреймворк Node.js), та для розробки нативних десктоп-програм (Electron). З 2018 року бібліотека React.js дозволяє створювати на своїй основі мобільні застосунки за допомогою React Native фреймворку.

Мова програмування JS має наступні переваги:

- швидкість;
- простота вивчення. JS інтуїтивно зрозуміла мова для вивчення програмістами-початківцями. Також відома своєю доступністю для «непрограмістів»;
- популярність. JS використовується у абсолютній більшості вебзастосунків, що значно спрощує розробку нових веббраузерів або нативних застосунків на основі JS фреймворків;
- дозволяє створювати добре маніпульовані та інтерактивні вебінтерфейси;
- підтримка великої кількості різноманітних бібліотек.

До недоліків мови можна віднести:

- по-різному інтерпретується основними веббраузерами: код, що працює в Google Chrome, може попереджати про помилки у Safari;
- відсутність типізації. Дуже незвична тема для більшості, навіть добре кваліфікованих програмістів, оскільки помилки, що виникають при неявному приведенні типів у JS, дуже важко відслідкувати помилки;
- класифікація JS як прототипної мови програмування. Розробники часто намагаються застосувати загальновідомі об'єктно-орієнтовні принципи програмування при написанні коду на JS. Але варто пам'ятати, що прототипна парадигма програмування різко відрізняється від об'єктно-орієнтовної, і потребує більш детального вивчення.

2.4.2. Adobe Flash

На початку 2000-х Flash був широко встановлений на настільних комп'ютерах, і його зазвичай використовували для відображення інтерактивних вебсторінок, онлайн-ігор та для відтворення відео та аудіоконтенту. У 2005 році YouTube був заснований колишніми співробітниками PayPal, і він використовував Flash Player як засіб для відображення стисненого відеовмісту в інтернеті.

У період з 2000 по 2010 рік численні компанії використовували вебсайти, що базуються на Flash, для запуску нових продуктів, або для створення інтерактивних порталів компаній. Найпопулярнішими користувачами були Nike, Hewlett-Packard, Nokia, General Electric, HBO, Cartoon Network, Disney та Motorola. Після того, як Adobe представила апаратно оптимізований 3D для Flash (Stage3D), вебсайти Flash відчули зростання 3D-вмісту для демонстрації продуктів та віртуальних турів.

Хоча Flash раніше був домінуючою платформою для онлайн-мультимедійного контенту, від нього поступово відмовляються, оскільки

Adobe виступає за перехід до HTML5, Unity чи інших платформ. Flash Player застарілий і має офіційний термін закінчення життя 31 грудня 2020 року. Однак Adobe продовжить розробляти Adobe Animate, який зосередиться на підтримці вебстандартів, таких як HTML5 замість застарілого формату Flash.

Тим не менш, Adobe Flash Player має наступні переваги:

- кросс-браузерна сумісність. Не потрібно турбуватися про різне трактування HTML в різних браузерах. При наявності Flash-плагіна, користувач зможе без проблем переглядати зміст вебсторінки;
- заміна зображення для спеціальних шрифтів. Flash можна використовувати для заміни текстових елементів на вебсторінках HTML на еквіваленти Flash;
- інтерактивність. Flash підтримує аудіо, анімацію та розширену обробку відео. Він заснований на принципах векторної графіки, що дозволяє відображати растрову графіку там, де це потрібно;
- якісна підтримка анімації.

До недоліків мови можна віднести:

- Flash потребує встановлення додаткового плагіну задля коректної роботи. При цьому його потрібно регулярно оновлювати, що значно ускладнює використання звичайних інтерактивних вебзастосунків;
- користувачі не можуть масштабувати розмір тексту вебзастосунку, що використовує Flash плагін;
- не будуть працювати вже вбудовані функції веббраузера (такі як кнопки «вперед» та «назад» для переходу від однієї вебсторінки до іншої), якщо не додати належного коду до Flash-застосунку;
- більшість мобільних пристроїв не в змозі коректно відображати Flash-застосунки;

- не дозволяє оптимізувати пошукові системи, оскільки Flash не надає змогу розбити ресурси застосунку на пошукові компоненти;
- довгий час завантаження Flash-застосунку;
- Flash-застосунки не відображаються у попередньому перегляді пошукових систем. Попередній перегляд дозволяє користувачеві краще зрозуміти, що очікувати на цільовому вебсайті перш ніж відвідувати його.

2.4.3. Microsoft Silverlight

Silverlight пропонує графічну систему, подібну до Windows Presentation Foundation (WPF), та інтегрує мультимедіа, графіку та анімацію у єдине середовище виконання. У додатках Silverlight користувальницькі інтерфейси розроблюються за допомогою розширюваної деларативної мови розмітки XAML, і програмуються за допомогою підмножини .NET Framework. XAML можна використовувати для розмітки векторної графіки та анімації.

Silverlight дозволяє динамічно завантажувати вміст розширюваної мови розмітки XML, яким можна керувати за допомогою інтерфейсу Document Object Model (DOM). Цей метод відповідає звичайному підходу до побудови користувацьких інтерфейсів. Silverlight також має змогу відкривати об'єкт Downloader, який використовується в тому числі і для завантаження потрібного застосунку вмісту (наприклад, сценаріїв, медіа-ресурсів або інших даних). Починаючи з версії 2.0 логіка програмування може бути написана будь-якою мовою .NET, в тому числі і загальними динамічними мовами програмування, такими як IronRuby або IronPython.

Одними з ключових переваг Silverlight є:

- застосунки, створені на основі Microsoft Silverlight, є кросс-платформенними, що надає перевагу не тільки користувачам, а і розробникам;

- багата підтримка дизайну інтерфейсу та швидка підтримка розробки застосунків;
- одними із підтримуваних мов програмування є Visual Basic, C#, Python та Ruby on Rails;
- підтримує 2D векторну графіку, відео високої роздільної здатності, аудіо та анімації;
- автоматичне оновлення;
- повністю читається сканерами пошукових систем;
- додаткові плагіни, потрібні для розробки того або іншого вебзастосунку на основі Microsoft Silverlight, займають мінімальну кількість системних ресурсів;
- легко інтегрується з .NET фреймворками.

До недоліків мови можна віднести:

- маленька частинка ринку;
- повільне завантаження файлів, які займають значний обсяг пам'яті. Silverlight не використовує метож стиснення зображень та текстових компонентів анімації;
- обмежена підтримка зображень. Silverlight повністю підтримує зображення з розширенням “.jpg” або “.png”. Може читати “.gif” формат, але лише в одно-, чотирьох-, восьми-бітових потоках даних. В свою чергу “.raw” розширення фреймворко Silverlight може лише читати, але ніяк не записувати;
- відсутність API для взаємодії з вебкамерою;
- складне розгортання. Silverlight розбиває вміст проекту на кілька файлів, на які потрібно посилатися окремо.

2.4.4. Результати аналізу

Врахувавши переваги та недоліки кожного із фреймворків, було вирішено, що в цій роботі буде доцільно використовувати Microsoft Silverlight як фреймворк для реалізації модуля клієнтської частини

застосунку, оскільки саме ця технологія найкраще підходить для розробки застосунку у середовищі Windows, а також має ряд необхідних бібліотек для швидкої та ефективної розробки функціональності вебзастосунку.

Крім того, Microsoft Silverlight відповідає вищеописаним загальним вимогам, що надає йому значну перевагу. Адже важлива саме інтеграція з середовищем розробки Microsoft Visual Studio, що забезпечить швидший процес розробки застосунку.

3. ОПИС СТРУКТУРО-АЛГОРИТМІЧНОЇ ОРГАНІЗАЦІЇ ВЕБЗАСТОСУНКУ

3.1. Загальний опис системи

Дана система – це вебзастосунок, який дає змогу користувачам створювати навчальний контент та об'єкти для навчальних платформ Learnsmart та Smartbook.

Неавторизований користувач не має доступу до даного застосунку, для використання повної функціональності продукту, таких як конвертація “.pdf” файлів, додавання, редагування та видалення навчальних об'єктів (LO), а також розгортання вже скомпільованих ресурсів адаптивного навчання на Learnsmart та Smartbook необхідно увійти в систему або зареєструватись. При початковому вході у систему користувач має змогу сконвертувати PDF-файл у відповідний “.bnn” формат, з яким вже може працювати LMS.

У даному вебзастосунку існує єдиний вид користувача. Особливою функціональністю, до якої має доступ кожен авторизований клієнт, є створення власного унікального онлайн-ресурсу, який відповідає інтересам та вимогам студентів (власне ті, заради кого і розміщуються відповідні навчальні об'єкти). При створенні такого навчального ресурсу заповнюється інформація про його назву, контактні дані авторів, а також для якого саме університету або курсу він призначений. Після успішної перевірки вже вбудованими інструментами LMS на наявність помилок компіляції, даний навчальний ресурс розміщується на відповідних навчальних платформах.

Для коректної взаємодії з навчальними ресурсами користувачеві надаються наступні можливості:

- конвертувати “.pdf” розширення файлу у “.bnn” формат;
- створювати, редагувати та видаляти власні навчальні ресурси;
- перевіряти їх на наявність помилок;

- збирати відповідну статистику про успішність студентів при проходженні даного LO;
- розміщувати інтелектуальні ресурси на навчальних платформах Learnsmart та Smartbook.

3.2 Опис вимог до функціональності вебзастосунку

В даному розділі описано основні функціональні вимоги до LMS. Зокрема, неавторизовані користувачі мають змогу:

- авторизуватися в системі;
- зареєструватися, застосовуючи спеціальну форму.

Вже авторизовані користувачі мають доступ до наступної функціональності:

- конвертація PDF-файлу у “.bnn” формат. Саме з цим розширенням вже може працювати постачальник навчального контенту. Незалежно від роздільної здатності вхідного файлу, конвертер завжди створюватиме вихідний файл з розширенням 200dpi;
- конвертація скопійованого ресурсу в формат EPUB;
- редагування інформації про назву ресурсу (дані автора, назва навчального закладу і т.д.). В кінці редагування дана інформація буде відображатись в загальній таблиці LO, тож користувач зможе повернутись та змінити бажані поля;
- додавання певних специфікацій до відповідної назви навчального ресурсу (титульне зображення, термін дії LO і т.д.).
- додавання модулів до кожного із навчальних ресурсів. Це створює певну ієрархію навчального курсу. Підказки вебзастосунку допоможуть користувачеві зорієнтуватися у використанні даної функціональності, але зазвичай на даному

етапі клієнт має змогу додати свої власні PDF-файли, або скопійовувати вже наявні у системі;

- видалення заголовку відповідного навчального ресурсу, або, власне, сам ресурс;
- доступ до двох видів маркувань: тегування LO та зображень;
- наявність функції “показати маркування”, за допомогою якої можна побачити теги LO. Також користувач матиме змогу відфільтрувати ці маркування за різними параметрами;
- створення проб для вибраних LO;
- ввімкнення функції Heat Map – вона забезпечує швидкий візуальний знімок використання студентами даної проби або тексту для відточення коректності навчального ресурсу;
- тестування контенту LO – за допомогою цієї функції користувач може перевірити на валідність кожен з окремих проб, або весь розділ цілком;
- можливість розміщувати вже готові навчальні ресурси на навчальних платформах.

3.3 Архітектура вебзастосунку

Для подальшої розробки даного вебзастосунку було обрано клієнт-серверну архітектуру.

Клієнт-серверна архітектура – це один із основних архітектурних шаблонів ПЗ. Ця концепція є найпоширенішою при створенні розподілених мережевих застосунків, вона забезпечує безперебійну взаємодію та обмін даними між ними.

Сервер може одночасно обслуговувати велику кількість клієнтів. Це дозволяє уникнути дублювання програмного коду як на стороні клієнта, так і на стороні сервера. Ба більше, через те, що всі дані зберігаються на сервері (точніше у БД, з якою вже спілкується безпосередньо сам сервер), а також через те, що всі операції та обчислення виконуються на тому ж

сервері, вимоги до комп'ютера, на якому встановлена та запущена клієнтська програма, суттєво знижуються.

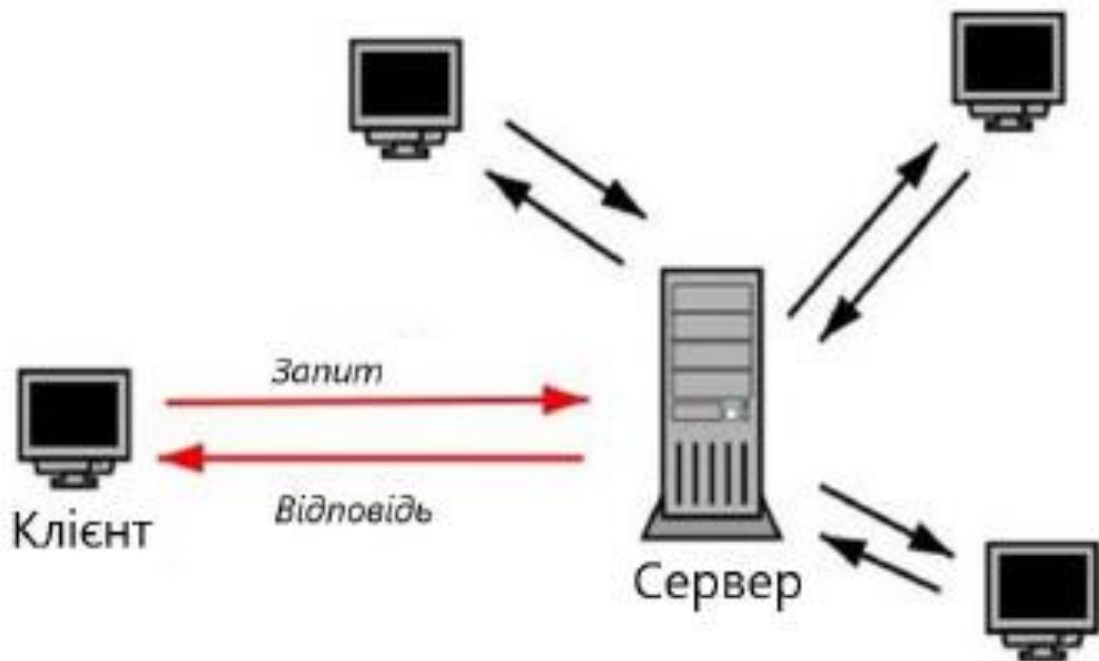


Рис. 1. Клієнт-серверна архітектура

Структура проєкту була розроблена на основі загальноприйнятих рекомендацій від авторів фреймворку Silverlight та .NET платформи. Також враховувалось розділення ПЗ на окремі системи функціональностей (модулі), та на окремі частини клієнтської частини застосунку (компоненти).

Розглянемо основні елементи для розробки користувацького інтерфейсу задля кращого розуміння структури вебзастосунку:

- ресурси (assets) – ця директорія відповідає за зберігання та обробку приватних даних користувача, а також складається з інтерфейсів, які відповідають за інтеграцію помилок та модулів безпеки у ПЗ;

- контролери (controllers) – кожен із контролерів є частиною відповідного компоненту, вони відповідають за відображення даних та за обробку запитів від шаблонів, до яких прив’язані;
- директиви (helpers) – створена для додавання власної функціональності до проєкту, а також зберігає деякі класи конвертації одного типу даних в інший;
- моделі (models) – описує загальні структури шаблонів, що використовуються для розробки клієнтської частини;
- шаблони (views) – є статичними частинами клієнтської частини, що відповідають за відображення окремих частин системи, функціонально виконують задачі рендерингу, підписки на події, використовують значення даних, які їм надає контролер, а також відображає зміну даних внаслідок роботи контролера. Є одним із основних компонентів фреймворку Silverlight.

Схема взаємодії відповідних компонентів фреймворку Silverlight представлена на рис. 2.

Загальна структура проєкту, яка складається з директорій, що відповідають за кожен із вищеописаних компонентів Silverlight, представлено на рис. 3.

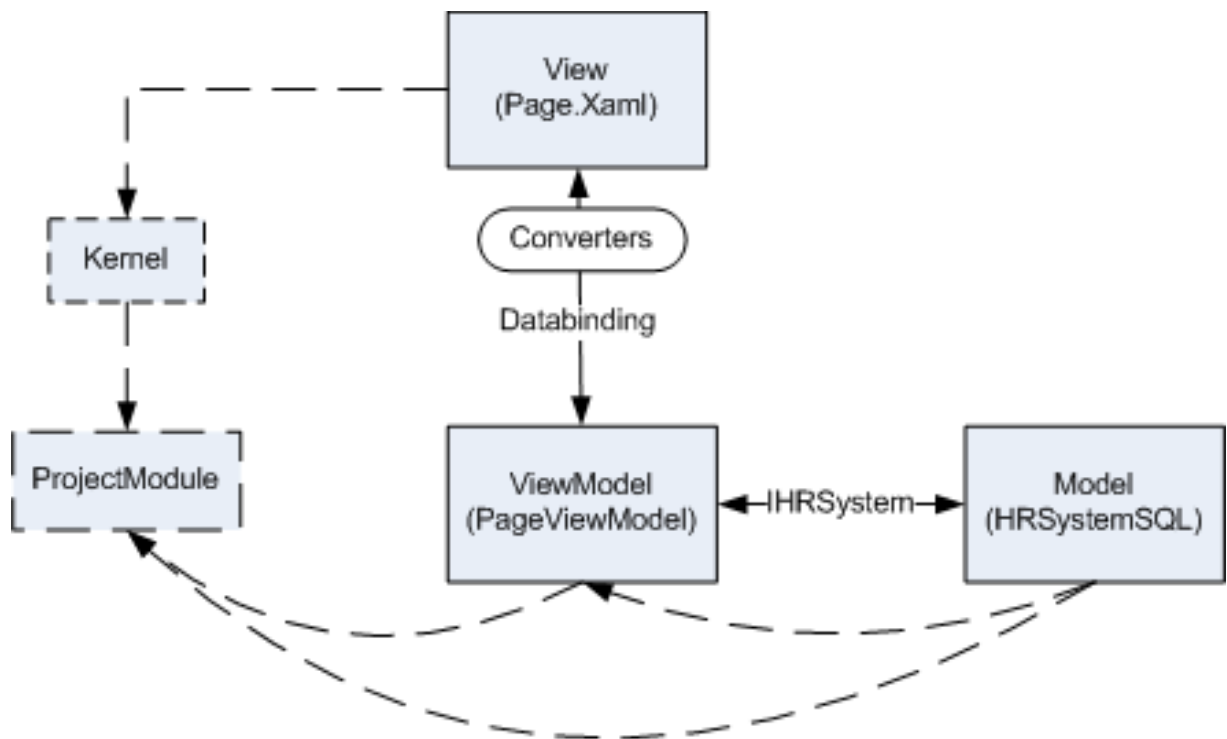


Рис. 2. Зв'язки між складовими Silverlight застосунку

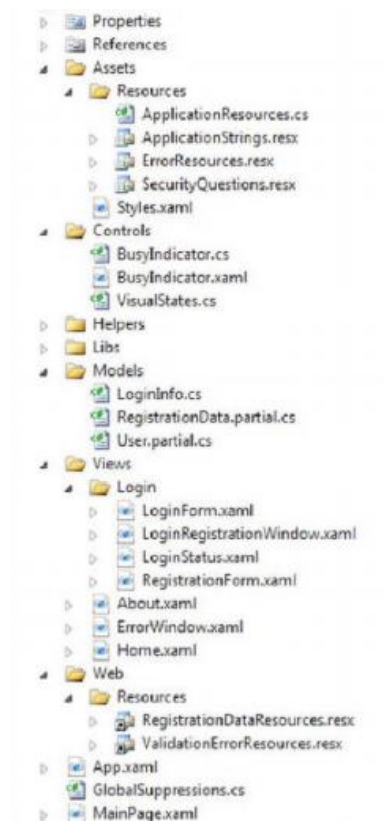


Рис. 3. Загальна структура клієнтської частини застосунку

3.4 Опис модулів ПЗ

При розробленні вебсервісу було виділено наступні модулі:

- модуль реєстрації та авторизації;
- модуль компонентів відображення;
- модуль зв'язку з БД;
- модуль компіляції та розміщення готових навчальних ресурсів.

Розглянемо найважливіші модулі розробленого вебзастосунку та їх взаємодію.

3.4.1. Модуль зв'язку з БД

Даний модуль надає змогу користувачу взаємодіяти з базою даних, та використовувати всю функціональність системи, передбачену для створення готових LO. Основною функцією цього модулю є оптимізація SQL-запитів. Усі дані, створені користувачем під час роботи з інтелектуальними ресурсами, такі як додавання проб, модулів та контенту LO, оброблюються в даному модулі, а вже після записуються у базу даних.

3.4.2. Модуль компіляції та розміщення готових LO

Даний модуль надає змогу користувачу зробити перевірку створених навчальних об'єктів. Він дозволяє оптимізувати рішення користувача щодо присвоєння проб відповідним частинам навчальних ресурсів, також перевіряє валідність заголовків та інших компонентів LO.

Після вдалого проходження вищезазначеного етапу користувач отримує змогу компілювати та розміщувати свої LO на навчальних платформах.

3.5 Висновки

В даному розділі було проведено детальний опис розроблюваної системи LMS, а також визначено функціональність, до якої має доступ авторизований або неавторизований користувач.

Було сформовано список вимог до функціональності вебзастосунку, що передбачає базовий набір функціональностей стандартної LMS, а також

враховано основні принципи роботи адаптивного навчання, та імплементовано загальні структурно-алгоритмічні вимоги.

Крім того, було розглянуто архітектуру ПЗ, яка найбільше підійде для розробки даної LMS, а саме клієнт-серверну архітектуру. Також враховувалось розділення ПЗ на окремі системи функціональностей (модулі), та на окремі частини клієнтської частини застосунку (компоненти).

В результаті було описано модулі, необхідні для повного функціонування розроблюваного ПЗ, та розглянуто основні з них.

.

4. АНАЛІЗ РОЗРОБЛЕНОГО ВЕБЗАСТОСУНКУ

Даний вебзастосунок було розроблено на основі стандартів шаблону MVC. Робочий потік застосунку було побудовано згідно з сучасними стандартами розроблення інтерфейсів вебзастосунків. Типова взаємодія користувача із розробленим вебресурсом починається із наступних кроків:

- вхід на сторінку вебресурсу після реєстрації або авторизації;
- завантаження основних модулів вебзастосунку;
- контролер, розроблюваний за допомогою фреймворку Microsoft Silverlight, відсилає запит на сервер задля отримання відповідних шаблонів, що вже надають змогу користувачеві взаємодіяти з функціональністю LMS;
- відображення шаблонів. Тепер клієнт може розпочати роботу, сконвертувавши PDF-файл у “.bnn” формат.

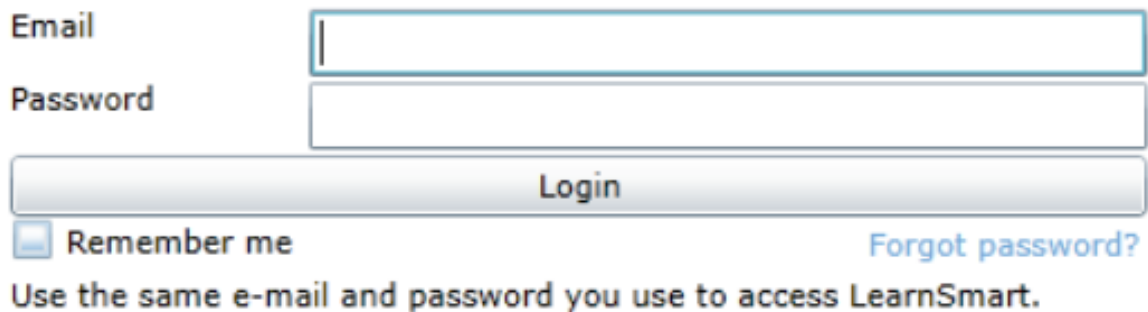
Після цього у користувача є можливість працювати вже з безпосередньо розробкою навчальних ресурсів для подальшого розміщення на навчальних платформах.

4.1. Робочий потік вебзастосунку

На головній сторінці вебзастосунку розміщена сторінка авторизації (рис. 4), яка складається з:

- поля для вводу електронної пошти користувача;
- замаскованого поля для вводу пароля;
- кнопки “Запам’ятати мене”, що дозволяє системі при повторній авторизації не запитувати від нього вводу електронної пошти та пароль;
- кнопки підтвердження форми.

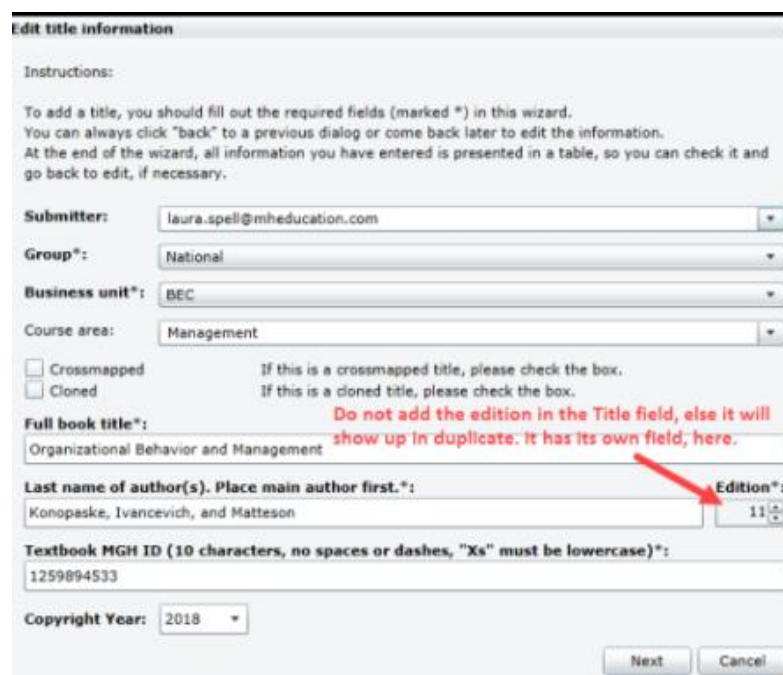
Після входу в систему, користувач має змогу споглядати завантаження відповідних шаблонів та ресурсів системи. Якщо завантаження пройшло успішно, клієнт отримує доступ до базової функціональності системи, який знаходиться у шапці вебзастосунку.



The image shows a user login interface. It includes a text input field for 'Email', another for 'Password', and a 'Login' button. Below the button is a checkbox labeled 'Remember me' and a blue link 'Forgot password?'. At the bottom, there is a note: 'Use the same e-mail and password you use to access LearnSmart.'

Рис. 4. Форма авторизації користувача

Після успішної конвертації PDF-файлу, користувач отримує можливість редагувати інформацію про назву ресурсу (рис. 5), що є необхідним кроком задля початку налаштування LO.



The image shows a 'Edit title information' form. It contains several dropdown menus: 'Submitter' (laura.spell@mheducation.com), 'Group*' (National), 'Business unit*' (BEC), and 'Course area' (Management). There are checkboxes for 'Crossmapped' and 'Cloned' with instructions. The 'Full book title*' field contains 'Organizational Behavior and Management'. A red arrow points to the 'Edition*' field, which has a value of '11'. A red text annotation says: 'Do not add the edition in the Title field, else it will show up in duplicate. It has its own field, here.' The 'Textbook MGH ID' field contains '1259894533' and the 'Copyright Year' is set to '2018'. 'Next' and 'Cancel' buttons are at the bottom right.

Рис.5. Форма заповнення інформації про заголовок LO

Після повного заповнення полів форми, користувач може підтвердити або відмінити її.

Поля, заповнювані користувачем:

- електронна пошта подавача навчального ресурсу. За замовчуванням тут буде вказана адреса користувача. Якщо право власності на проєкт зміниться, потрібно буде ввести електронну пошту іншої відповідальної людини для того, щоб вона могла в подальшому компілювати та розгортати на навчальних платформах даний LO;
- група. Це поле відповідає за вибір сфери поширення даного навчального ресурсу: міжнародний, національний і т.д.;
- зона поширення курсу. Тут конкретизується, для якого саме навчального закладу призначений даний LO;
- повна назва навчального ресурсу. Якщо LO користувача є міжнародним, варто вказати у заголовку певне слово, за допомогою якого даний ресурс можна буде відрізнити від національного;
- ім'я та прізвища авторів;
- номер видання;
- ідентифікатор ресурсу. Це автоматично згенерована строка, яка повинна прийти на пошту користувача задля безпеки. В будь-якому випадку, в даному полі можна ввести несправжній ідентифікатор, і вже з часом отримати його через відповідні джерела.

Одразу після проходження вищеописаного етапу, клієнт отримує змогу промаркувати частини навчального ресурсу (в основному текст, хоча буває і медіа, як наприклад аудіо, відео, картинки і т.д.) для подальшої роботи з пробами. На основі даних тегувань (рис. 6) вже можна вибудувати структуру проб, які безпосередньо відображають основний принцип

адаптивного навчання – учень сам налаштовує навчальний процес, незалежно від викладача.

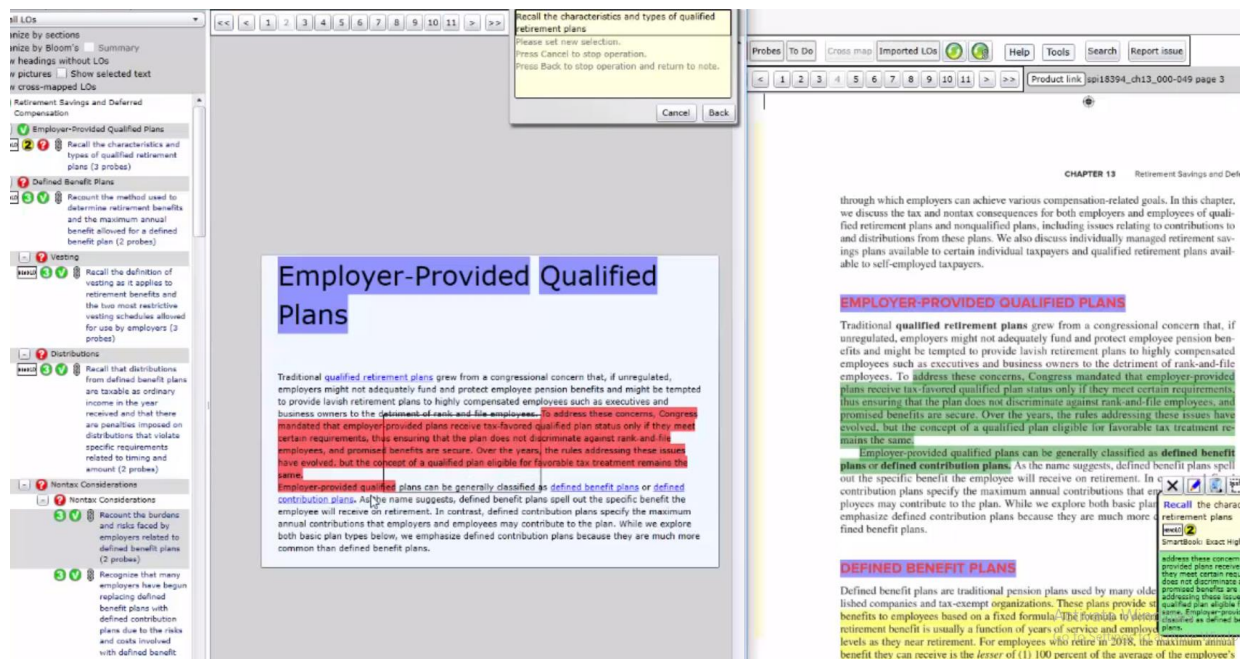


Рис. 6. Тагування навчального ресурсу

Наступним кроком до створення повноцінного LO є присвоєння проб супровідним маркованим частинам навчального ресурсу. Під час створення проб, які вимагають від учня використати набуті знання для вирішення проблеми, автор повинен керуватися наступними вказівками:

- уникати використання завдання, що вже вирішено у тексті навчального ресурсу;
- намагатися побудувати задачі, які студент повинен вирішити застосовуючи основні поняття та прийоми, що змодельовані в тексті;
- задача не повинна бути складною. Вона має забезпечувати студенту можливість використати вивчені навички або поняття задля самооцінки освоєння цих самих навичок або понять. Адаптивне навчання не бере за мету замінити звичайні домашні завдання, які не надають такої можливості.

Щоб відкрити вікно для створення проб, потрібно натиснути на кнопку “Проби” у верхньому меню (рис. 7). Відповідно у відображеному списку вже можна обрати тип проби.

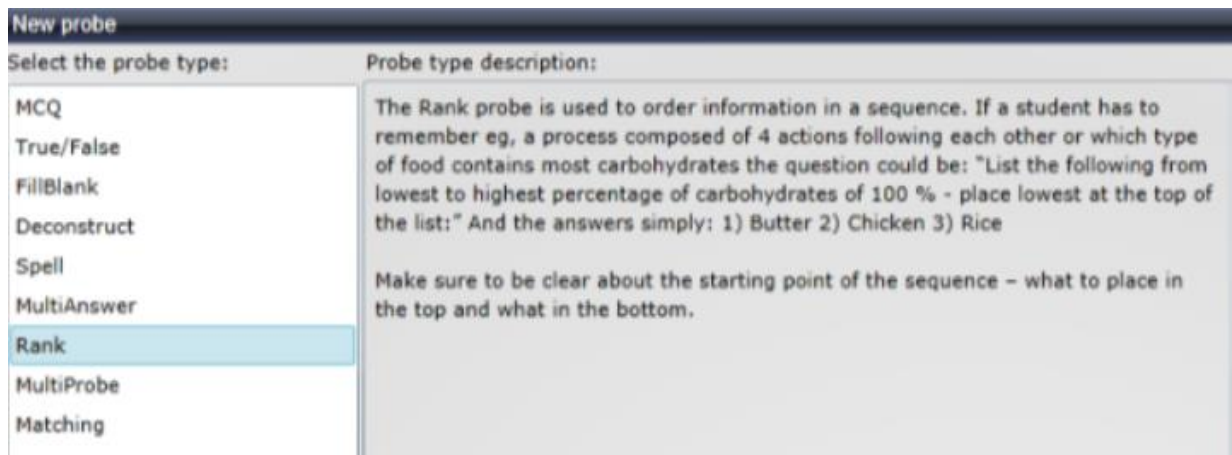


Рис. 7. Вибір проб

Після присвоєння всіх проб, користувач отримує змогу перевірити свій навчальний ресурс на валідність, і лиш після успішної перевірки – розміщувати даний LO на навчальних платформах.

4.2. Тестування вебзастосунку

Контроль за якістю є постійною та важливою частиною життєвого циклу розроблення та підтримки програмного забезпечення та входить в кожен етап створення вебзастосунку, як завершальна частина перед затвердженням імплементації функціональності.

Тестування направлено на виявлення помилок наступних типів:

- інтерфейсних (помилки стилізації, відображення сторінок);
- функціональних (перевірки правильності роботи частин застосунку та інформаційних потоків);
- авторизаційних (захищеність від доступу до окремих елементів інтерфейсу та конфіденційних даних для незареєстрованих або з недостатнім рівнем доступу користувачів).

Одним із способів тестування ПЗ було ручне тестування, за допомогою якого вводилися різні дані до текстових полів, зокрема

некоректні дані, пусті значення тощо. За результатами цього тестування було додано повідомлення для користувачів про відповідні помилки, які виникають при введенні ними некоректних даних.

4.3. Порівняння розробки із існуючими аналогами

Розроблений вебзастосунок відповідає усім визначеним вимогам, а саме:

- можливість конвертації PDF-файлу у визначений “.bnp” формат, з яким вже безпосередньо працює LMS;
- можливість додавання інформації про заголовки того чи іншого ресурсу задля подальшого розміщення LO на навчальних платформах;
- можливість маркування частин тексту;
- можливість додавання проб;
- можливість розміщення інтелектуальних ресурсів на навчальних платформах.

Окрім цього було усунуто наступні недоліки альтернативних програмних засобів

- вузька направленість аналогів;
- відображення контенту лише для певної території;
- складність створення та просування власних навчальних ресурсів;
- складність тегування та маркування LO;
- складний та неінтуїтивний інтерфейс.

4.4. Рекомендації щодо подальшого вдосконалення

Розроблений вебзастосунок можна застосовувати для створення інтерактивних навчальних ресурсів, оскільки застосунок надає достатньо функціональних можливостей для створення та керування пробами.

Отже, розроблене програмне забезпечення можна і потрібно вдосконалювати, розширюючи його функціональність для подальшого

використання для просування навчальних ресурсів локальних організаторів та великих комерційних компаній, що прагнуть зробити популярнішими LMS.

Для цього можна надати наступні можливості:

- додавання можливості маркувати не тільки текст, а і аудіо, відео або зображення;
- додавання можливості складати звіти на основі успішності проходження учнями відповідних проб;
- інтеграція сервісу як частини популярних навчальних платформ;
- розширення аналізу аудиторії сайту;
- реалізація багатомовного інтерфейсу;
- додання системи сповіщень електронною поштою для користувачів на різноманітні події;
- додавання системи підписки на профіль автора;
- розробка альтернативних версій застосунку для різноманітних ОС.

4.5. Висновки

В даному розділі було проведено детальний опис розроблюваної системи LMS, а також визначено робочий потік застосунку, до якої має доступ авторизований або неавторизований користувач.

Було сформовано список вимог до тестування функціональності вебзастосунку, що передбачає базовий набір тестування функціональностей стандартної LMS. Враховано основні принципи роботи адаптивного навчання, та імплементовано загальні структурно-алгоритмічні вимоги.

Крім того, було розглянуто функціональність альтернативних LMS. Було здійснено порівняння вже наявної функціональності системи з - положення, які надають перевагу розробленій сисетмі.

В результаті було описано рекомендації, необхідні для повноцінного функціонування розроблюваного ПЗ, та розглянуто основні з них.

Крім того, було розглянуто архітектуру ПЗ, яка найбільше підійде для розробки даної LMS, а саме клієнт-серверну архітектуру. Також враховувалось розділення ПЗ на окремі системи функціональностей (модулі), та на окремі частини клієнтської частини застосунку (компоненти).

В результаті було описано модулі, необхідні для повного функціонування розроблюваного ПЗ, та розглянуто основні з них.

ВИСНОВКИ

У першій частині цієї роботи наведено загальний опис роботи LMS та їх призначення. Проаналізовано принцип роботи даного програмного застосунку. Також обґрунтовано проблеми, пов'язані з використанням LMS. Також було наведено існуючі реалізації методів адаптивного навчання: Chamilo, Claroline, eFront, обґрунтовано їх складність та що кожен із цих програмних застосунків по-різному підходить до вирішення головної задачі. Крім того, було сформовано вимоги до даної дипломної роботи. Вимоги стосуються програмної та логічної частин.

У другому розділі було розглянуто відповідні засоби розробки, необхідні для грамотної імплементації функціональності системи. Було вирішено, що в цій роботі буде доцільно використовувати вебзастосунок як вид ПЗ, оскільки саме цей тип більш гнучкий у виборі технологій розробки, має менші витрати на розробку програмного коду, а також вебзастосунки легше оновлювати при виявленні помилок.

Також було проаналізовано RDBMS Microsoft SQL Server та MySQL, порівнявши їх переваги недоліки, було виділено наступні пріоритети SQL Server над MySQL:

- відмінно структуроване та інтуїтивно зрозуміле ПЗ у SQL Server для управління БД. Лиш кілька конкурентів (одним із них є MySQL) змогли розробити подібне ПЗ, але SQL Server Enterprise Edition є набагато простішим у використанні та має більше функцій;
- повна підтримка тригерів у SQL Server на відміну від MySQL, який лиш недавно презентував їх. Тим не менш, вони не підтримуються повністю;

- налагоджений процес відновлення даних на відміну від MySQL, який схильний до неефективної обробки транзакцій, що призводить до пошкодження даних;
- Microsoft Visual Studio має вбудовану підтримку конфігурації Microsoft SQL Server та програмування специфічних для цієї RDBMS SQL запитів.

В останніх розділах було проведено детальний опис розроблюваної системи LMS, а також визначено функціональність, до якої має доступ авторизований або неавторизований користувач. Було сформовано список вимог до функціональності вебзастосунку, що передбачає базовий набір функціональностей стандартної LMS. Враховано основні принципи роботи адаптивного навчання, та імплементовано загальні структурно-алгоритмічні вимоги.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

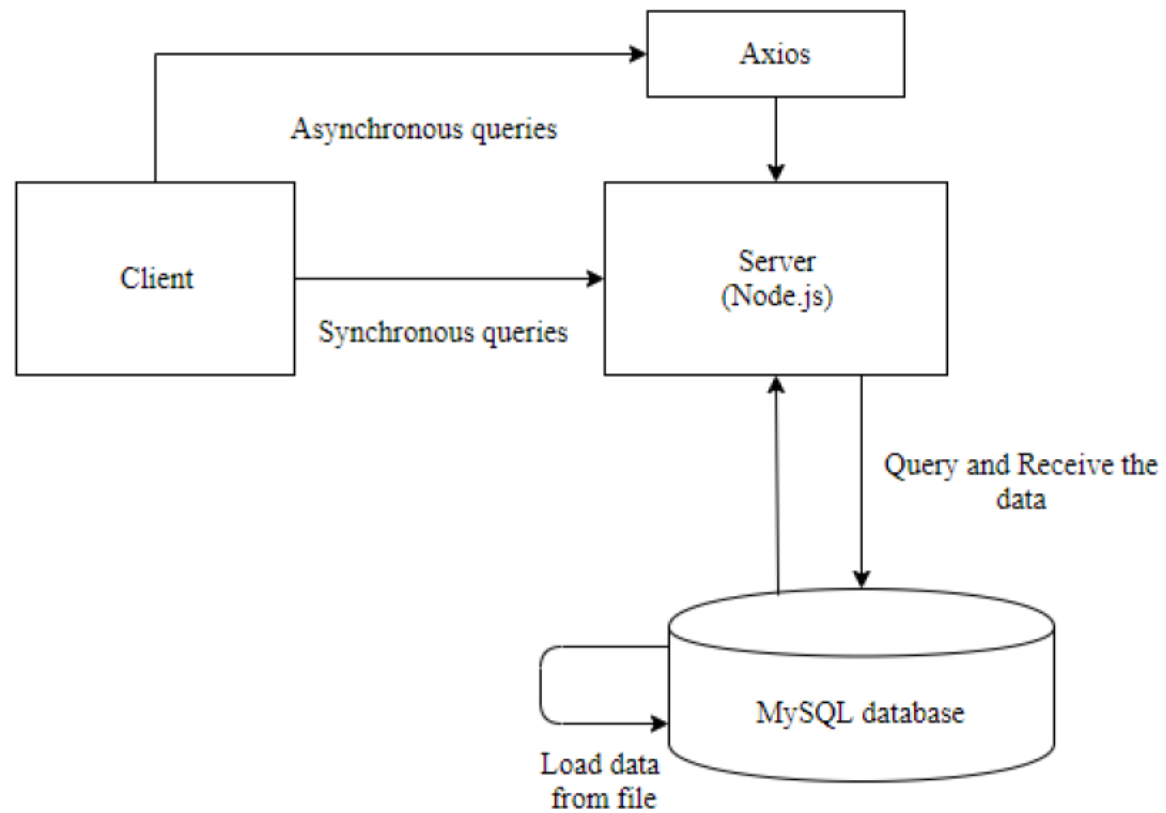
1. To Raise Productivity, Let More Employees Work from Home : [Електронний ресурс]. Режим доступу: <https://hbr.org/2014/01/to-raise-productivity-let-more-employees-work-from-home>
2. Harvest: Simple Online Time Tracking Software : [Електронний ресурс]. Режим доступу: <https://www.getharvest.com/>
3. Jira Software : [Електронний ресурс]. Режим доступу: <https://www.atlassian.com/ru/software/jira>
4. PYPL PopularitY of Programming Language [Електронний ресурс] — Режим доступу: <http://pypl.github.io/PYPL.html>
5. TIOBE Index for May 2020 [Електронний ресурс] — Режим доступу: <https://www.tiobe.com/tiobe-index/>
6. The RDBMS Rankings: January 2020 [Електронний ресурс] — Режим доступу: <https://redmonk.com/sograzy/2020/02/28/language-rankings-1-20/>
7. The Top Programming Languages 2019 [Електронний ресурс] — Режим доступу: <https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019>
8. About Microsoft Silverlight : [Електронний ресурс]. Режим доступу: <https://silverlight.org/en/about/>
9. Янг, Алекс. .NET в действии [Текст] / Алекс Янг, Брэдлі Мек, Майк Кантелон. — СПб. : Питер, 2015. — 448 с.
10. Express : [Електронний ресурс]. Режим доступу: <https://expressjs.com/>
11. About Chamilo : [Електронний ресурс]. Режим доступу: <https://www.chamilo-lang.org/en/about/>
12. eFront: [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/eFront>
13. Python Reference Manual : [Електронний ресурс]. Режим доступу: <https://docs.python.org/2.4/ref/front.html>

14. What Is LMS? : [Электронный ресурс]. Режим доступа: <https://en.lms.org/tutorial/tutorial.html#what-is-react>
15. Introduction to Angular concepts : [Электронный ресурс]. Режим доступа: <https://angular.io/guide/architecture>
16. Vue.js in Action [Текст] / E. Hanchett, B. Listwon. — Shelter Island : Manning, 2018. — 304 p.
17. DB-Engines Ranking : [Электронный ресурс]. Режим доступа: <https://db-engines.com/en/ranking>
18. High Performance Microsoft SQL Server, 3rd Edition [Текст] / Baron Schwartz, Peter Zaitsev, Vadim Tkachenko. — Sebastopol : O'Reilly Media, 2012. — 864 с.
19. Introduction to Redis : [Электронный ресурс]. Режим доступа: <https://redis.io/topics/introduction>
20. MySQL: The Definitive Guide Second Edition [Текст] / Kristina Chodorow. — Sebastopol : O'Reilly Media, 2013. — 432 p.
21. Model-View-Controller в .Net : [Электронный ресурс]. Режим доступа: <http://rsdn.org/article/patterns/modelviewpresenter.xml>
22. Model-View-ViewModel : [Электронный ресурс]. Режим доступа: <https://uk.wikipedia.org/wiki/Model-View-ViewModel>
23. ApexChartsJS : [Электронный ресурс]. Режим доступа: <https://apexcharts.com/>
24. What is Manual Testing? : [Электронный ресурс]. Режим доступа: <http://www.softwaretestingclass.com/what-is-manual-testing/>

ДОДАТКИ

Додаток 1

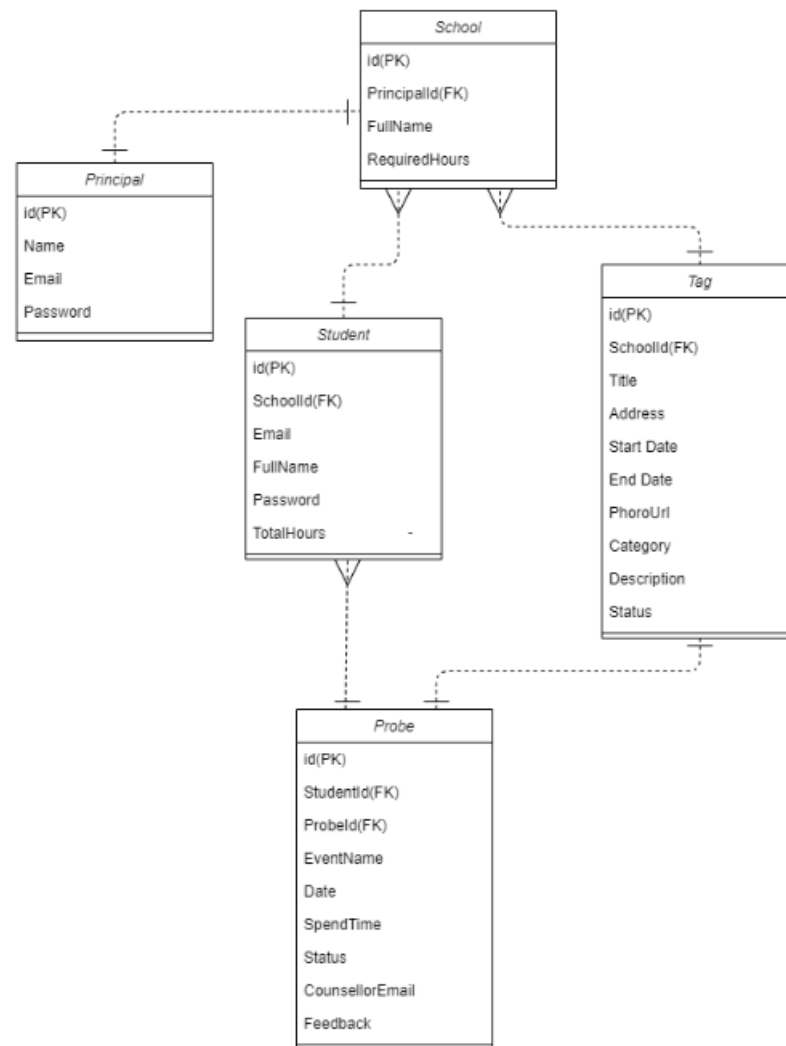
Копії графічних матеріалів



ДП.045440-06-99.

Постачальник контенту для навчальних платформ.

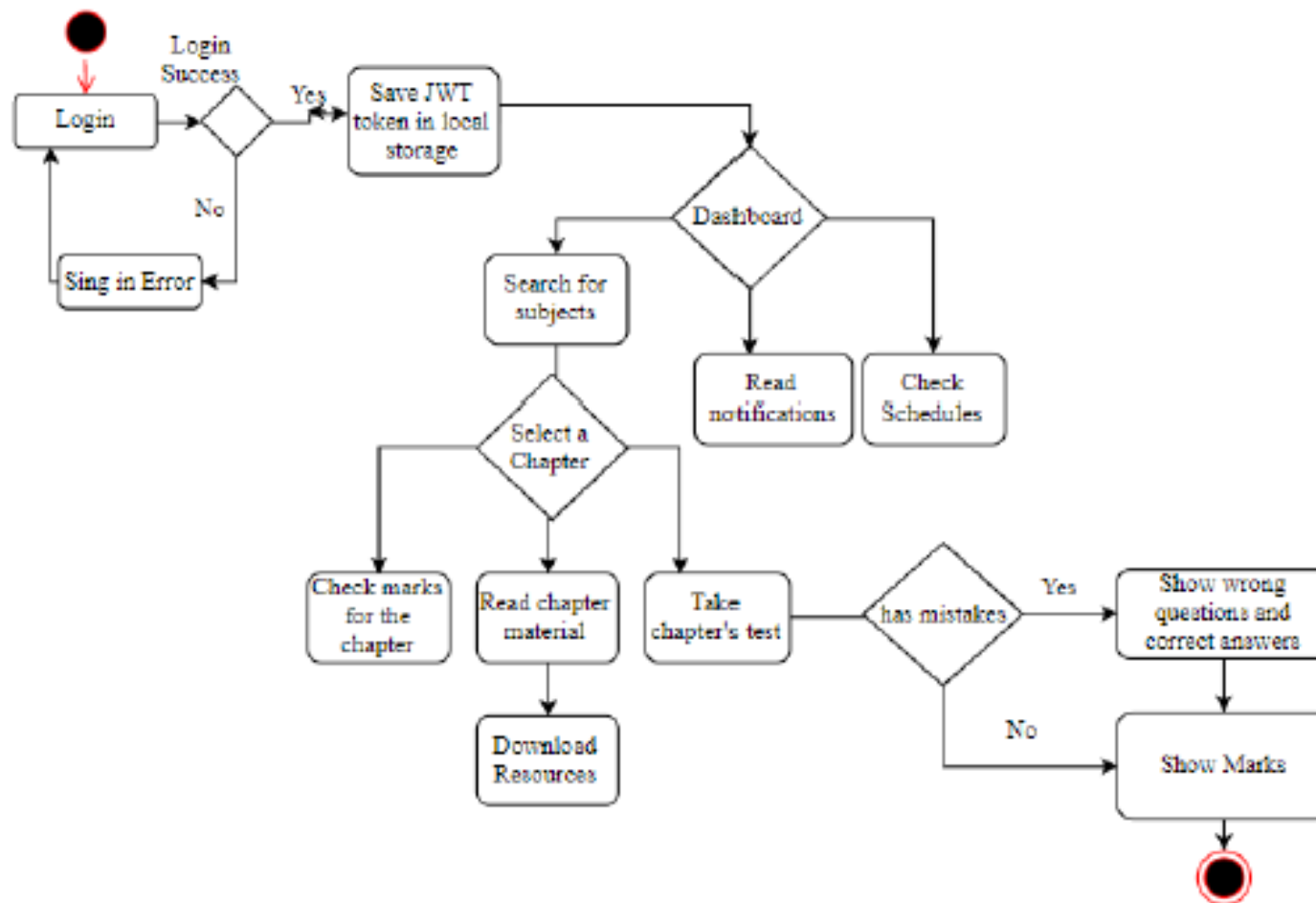
Архітектура системи. Компоненти платформи



ДП.045440-07-99.

Постачальник контенту для навчальних платформ.

Архітектура системи. Схема бази даних
платформи. ER-діаграма



Благодатний Дмитро Романович
гр. КП-62

Convert pdf tool (ver. 1.60)

☒ Show all titles [Change Filter](#) [Tools](#) [Help](#) [No hide](#) ▼ Titles: 1/0 ☐ Collapse all [Group filter](#) [Business unit filter](#) [Flag filter](#) [Extra](#)

	Authors	ISBN	Product ID	Course	Business unit	Group	Region	Compile	Status	Submitter
	slkdcmlksdnc, 1e	1277273999			HSSL	National	n/a	Compile	New	Dmytro.Blahodatny

--

Heading	LOs	Probes in total	Probes local only	Local probes in % of total	Compile	Content	Deadline	Assignment	Time spent last 7 days (except today)	Start page	Pages count	Images	Export	Crop	
---------	-----	-----------------	-------------------	----------------------------	---------	---------	----------	------------	---------------------------------------	------------	-------------	--------	--------	------	--

Благодатний Дмитро Романович
гр. КП-62

Додаток 2
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

Постачальник контенту для навчальних платформ Learnsmart та Smartbook

Виконав: Благодатний Дмитро Романович

Науковий керівник: старший викладач каф. ПЗКС ФПМ, Бухтіяров Юрій
Вікторович

Activate W
Go to Settings

Київ – 2020



ПОСТАНОВКА ЗАДАЧІ

Мета проекту: розробити програмну платформу, за допомогою якої користувач матиме змогу розробляти та розміщувати навчальні ресурси на Learnsmart та Smartbook платформах.

Завдання:

1. Проаналізувати загальні методи адаптивного навчання та програмні реалізації цих методів.
2. Розробити веб-інтерфейс.
3. Розробити алгоритм адаптивного навчання
4. Протестувати роботу системи



АКТУАЛЬНІСТЬ

- Адаптивне навчання стрімко набирає обертів у зв'язку з активно прогресуючим онлайн-навчанням (e-Learning).
- Процес створення унікальної LMS (Системи Керування Навчанням) займає багато часу. Від початкових макетів до створення мінімально працюючого прототипу проходить близько трьох місяців.
- Розробка алгоритму адаптивного навчання, який буде чітко відповідати вимогам студентів та викладачів, достатньо ресурсоємка задача.



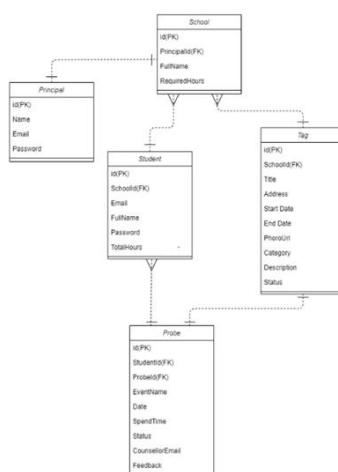
ВИМОГИ

- Система повинна мати конструктор ресурсів навчання (такі як лабораторні, практики, лекції і т.д.).
- Інтеграція з навчальними платформами Learnsmart та Smartbook.
- Автоматична компіляція та генерація ресурсів на основі просяаних інструкцій
- Публікація підтверджених ресурсів до навчальних платформ.



АРХІТЕКТУРА СИСТЕМИ

Загальна структура системи:

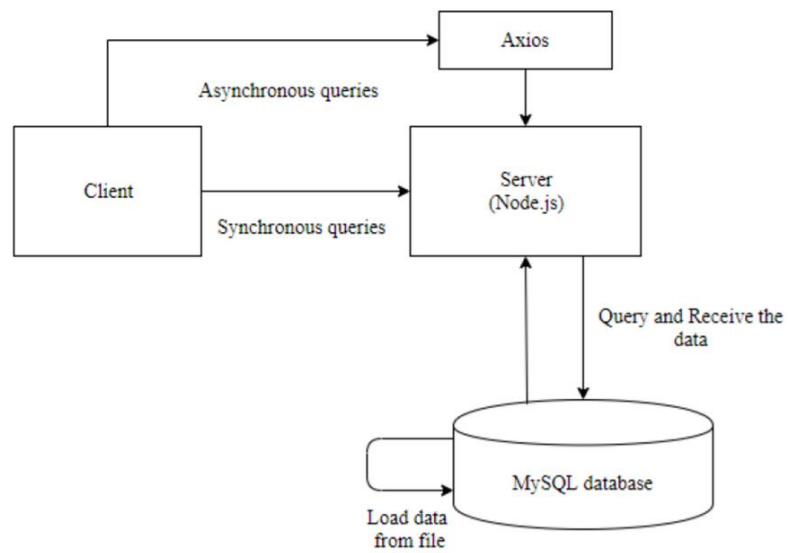




АРХІТЕКТУРА СИСТЕМИ

- Керуючий сервіс системи
- Мікросервіс компіляції контенту
- Мікросервіс попереднього рендерингу контенту в HTML формат
- Мікросервіс розміщення контенту на відповідних платформах
- Взаємодія мікросервісів

Бази даних





ЗАСОБИ РОЗРОБЛЕННЯ

- Мова програмування – C#
- Програмна платформа - .NET
- Бази даних: Microsoft SQL Server
- Веб-інтерфейс: Microsoft Silverlight



ВИМОГИ ДО ІНТЕРФЕЙСУ

- Інтерфейс системи мати інтуїтивно зрозумілі інтерактивні елементи а також забезпечувати просту систему пошуку та фільтрації контенту.
- Елементи інтерфейсу повинні бути однаково зрозумілими користувачам з різних мовних регіонів.
- Інтерфейс системи повинен бути розрахований на користувачів, які не мають спеціальних технічних знань і навичок в області комп'ютерної техніки.



ВИМОГИ ДО ПРОДУКТУ

- Забезпечення зручності, якості, швидкого розуміння роботи з програмою
- Можливість генерувати контент з відповідними полями (заголовками, пробами)
- Поля у заблокованому режимі поєднуються і вважаються одним полем
- Поля у розблокованому режимі утворюють простір усіх можливих комбінацій
- Перевірка утворених комбінацій з існуючими варіаціями у системі, знаходження їх різниць
- Оновлення вже існуючих варіацій новими, на основі отриманих даних про зміни, додавання нових варіацій до системи та видалення застарілих

ПРИКЛАД РОБОТИ ПРОГРАМИ



ПРИКЛАД РОБОТИ ПРОГРАМИ

Phases of Cellular Respiration

Cellular respiration involves four phases: glycolysis, the preparatory reaction, the citric acid cycle, and the electron transport chain (Fig. 8.2). Glycolysis takes place outside the mitochondria and does not require the presence of oxygen. Therefore, glycolysis is **anaerobic**. The other phases of cellular respiration take place inside the mitochondria, where oxygen is the final acceptor of electrons. Because they require oxygen, these phases are called **aerobic**.

During these phases, notice where CO_2 and H_2O , the end products of cellular respiration, and ATP, the main outcome of respiration, are produced.

Glycolysis (Gk: glycos, sugar, and lytic, splitting) is the breakdown of glucose (a 6-carbon molecule) to two molecules of pyruvate (two 3-carbon molecules). Oxidation results in NADH and provides enough energy for the net gain of two ATP molecules.

The **preparatory (prep) reaction** takes place in the matrix of the mitochondrion. Pyruvate is broken down from a 3-carbon (C_3) to a 2-carbon (C_2) acetyl group, and a 1-carbon CO_2 molecule is released. Since glycolysis ends with two molecules of pyruvate, the prep reaction occurs twice per glucose molecule.

with a 4-carbon molecule, forming two 6-carbon citrate molecules. As citrate bonds are broken and oxidation occurs, NADH and FADH₂ are formed, and two CO_2 per citrate are released. The citric acid cycle is able to produce one ATP per turn. Because two acetyl groups enter the cycle per glucose molecule, the cycle turns twice.

The **electron transport chain (ETC)** is a series of carriers in the cristae of the mitochondrion. NADH and FADH₂ give up their high-energy electrons to the chain. Energy is released and captured as the electrons move from a higher-energy to a lower-energy state during each redox reaction. Later, this energy is used for the production of ATP by chemiosmosis. After oxygen receives electrons at the end of the chain, it combines with hydrogen ions (H^+) and becomes water (H_2O).

Pyruvate, the end product of glycolysis, is a pivotal metabolite; its further treatment depends on whether oxygen is available. If oxygen is available, pyruvate enters a mitochondrion and is broken down completely to CO_2 and H_2O as shown in the cellular respiration equation (p. 136). If oxygen is not available, pyruvate is further metabolized in the cytoplasm by an anaerobic process called **fermentation**. Fermentation results in a net gain of only two ATP per glucose molecule.

Check Your Progress

1. Explain the benefit of slow glucose breakdown rather than rapid breakdown during cellular respiration.
2. Describe the four phases of complete glucose breakdown, including which release CO_2 and which produce H_2O .

8.1

- Red** – 60% or fewer students answered it correctly
- Orange** – between 60-70% of students answered it correctly
- Yellow** – between 70-80% of students answered it correctly
- Green** – More than 80% of students answered it correctly

РЕЗУЛЬТАТИ



- Розроблений продукт є унікальним та не містить аналогів
- Усі вимоги були виконані у повному обсязі
- Повна автоматизація створення та компіляції навчальних ресурсів



ВИСНОВКИ

1. Проаналізовано методи адаптивного навчання для різних навчальних платформ.
2. Розроблено мікросервісну архітектуру ПЗ.
3. Інтегровано платформу з основним функціоналом LMS.
4. Розроблено структуру баз даних.
5. Розроблено веб-інтерфейс для роботи з системою.
6. Протестовано програмні засоби.

ПЕРЕВІРКА НА АНТИПЛАГІАТ



Частина пояснювальної записки	Унікальність
Вступ	98%
Розділ 1	93%
Розділ 2	91%
Розділ 3	98%
Розділ 4	98%
ПЗ	93%



Дякую за увагу!

Додаток 3
Текст програми

```

import React from 'react';
import PropTypes from 'prop-types';
import { makeStyles } from '@material-ui/core/styles';
import Table from '@material-ui/core/Table';
import TableBody from '@material-ui/core/TableBody';
import TableCell from '@material-ui/core/TableCell';
import TableContainer from '@material-ui/core/TableContainer';
import TableHead from '@material-ui/core/TableHead';
import TableRow from '@material-ui/core/TableRow';
import TableSortLabel from '@material-ui/core/TableSortLabel';
import Typography from '@material-ui/core/Typography';
import Paper from '@material-ui/core/Paper';
import IconButton from '@material-ui/core/IconButton';
import KeyboardArrowDownIcon from '@material-ui/icons/KeyboardArrowDown';
import KeyboardArrowUpIcon from '@material-ui/icons/KeyboardArrowUp';
import Collapse from '@material-ui/core/Collapse/Collapse';
import Box from '@material-ui/core/Box';

function descendingComparator(a, b, orderBy) {
  if (b[orderBy] < a[orderBy]) {
    return -1;
  }
  if (b[orderBy] > a[orderBy]) {
    return 1;
  }
  return 0;
}

function getComparator(order, orderBy) {
  return order === 'desc'
    ? (a, b) => descendingComparator(a, b, orderBy)
    : (a, b) => -descendingComparator(a, b, orderBy);
}

function stableSort(array, comparator) {
  const stabilizedThis = array.map((el, index) => [el, index]);
  stabilizedThis.sort((a, b) => {
    const order = comparator(a[0], b[0]);
    if (order !== 0) return order;
    return a[1] - b[1];
  });
  return stabilizedThis.map((el) => el[0]);
}

const headEventCells = [
  { id: 'name', numeric: false, disablePadding: true, label: 'Event Name' },
  { id: 'rate', numeric: true, disablePadding: false, label: 'Event Rate' },
  { id: 'start', numeric: true, disablePadding: false, label: 'Start Date' },
  { id: 'marksCount', numeric: true, disablePadding: false, label: 'Count of time tracks' },
];

const headCategoriesCells = [
  { id: 'name', numeric: false, disablePadding: true, label: 'Category Name' },
  { id: 'rate', numeric: true, disablePadding: false, label: 'Category Rate' },
  { id: 'eventsCount', numeric: true, disablePadding: false, label: 'Count of Category events' },
];

const headStudentCells = [

```

```

    { id: 'name', numeric: false, disablePadding: true, label: 'Student Name'
  },
  { id: 'rate', numeric: true, disablePadding: false, label: 'Student Rate'
  },
  { id: 'time', numeric: true, disablePadding: false, label: 'Student
Approved Time' },
  { id: 'marksCount', numeric: true, disablePadding: false, label: 'Count
of time tracks' },
];

```

```

function EnhancedTableHead(props) {
  const { classes, order, orderBy, onRequestSort, tableType } = props;
  const createSortHandler = (property) => (event) => {
    onRequestSort(event, property);
  };
  let headCells = [];
  switch (tableType) {
    case 'events':
      headCells = headEventCells;
      break;
    case 'categories':
      headCells = headCategoriesCells;
      break;
    case 'students':
      headCells = headStudentCells;
      break;
  }

  return (
    <TableHead>
      <TableRow>
        <TableCell padding="checkbox">
        </TableCell>
        {headCells.map((headCell) => (
          <TableCell
            key={headCell.id}
            align={headCell.numeric ? 'right' : 'left'}
            padding={headCell.disablePadding ? 'none' :
'default'}
            sortDirection={orderBy === headCell.id ? order :
false}
          >
            <TableSortLabel
              active={orderBy === headCell.id}
              direction={orderBy === headCell.id ? order :
'asc'}
              onClick={createSortHandler(headCell.id)}
            >
              {headCell.label}
              {orderBy === headCell.id ? (
                <span className={classes.visuallyHidden}>
ascending'>
                  {order === 'desc' ? 'sorted descending' : 'sorted
ascending'}
                </span>
              ) : null}
            </TableSortLabel>
          </TableCell>
        ))}
      </TableRow>
    </TableHead>
  );
}

```

```

EnhancedTableHead.propTypes = {

```



```

    classes: PropTypes.object.isRequired,
    numSelected: PropTypes.number.isRequired,
    onRequestSort: PropTypes.func.isRequired,
    onSelectAllClick: PropTypes.func.isRequired,
    order: PropTypes.oneOf(['asc', 'desc']).isRequired,
    orderBy: PropTypes.string.isRequired,
    rowCount: PropTypes.number.isRequired,
  };

const useStyles = makeStyles((theme) => ({
  root: {
    width: '100%',
  },
  paper: {
    width: '100%',
    marginBottom: theme.spacing(2),
  },
  table: {
    minWidth: 750,
  },
  visuallyHidden: {
    border: 0,
    clip: 'rect(0 0 0 0)',
    height: 1,
    margin: -1,
    overflow: 'hidden',
    padding: 0,
    position: 'absolute',
    top: 20,
    width: 1,
  },
}));

function EventTableBody({rows, isSelected, order, orderBy}) {
  const [open, setOpen] = React.useState({});
  return (
    <TableBody>
      {stableSort(rows, getComparator(order, orderBy))
        .map((row, index) => {
          const isSelected = isSelected(row.name);
          const labelId = `enhanced-table-checkbox-${index}`;

          return (
            <React.Fragment key={index}>
              <TableRow
                hover
                aria-checked={isSelected}
                tabIndex={-1}
                key={row.name}
                selected={isSelected}
              >
                <TableCell component="th" scope="row">
                  <IconButton aria-label="expand row"
                    size="small" disabled={row.marksCount === 0} onClick={() => {
                      let newOpen = {...open};
                      newOpen[`_${index}`] =
!newOpen[`_${index}`];

                      setOpen({...newOpen})
                    }}>
                    {open[`_${index}`] ?
<KeyboardArrowUpIcon /> : <KeyboardArrowDownIcon />}
                  </IconButton>
                </TableCell>

```

```

scope="row" padding="none">
    <TableCell component="th" id={labelId}
        {row.name}
    </TableCell>
    <TableCell
align="right">{row.rate}</TableCell>
    <TableCell align="right">{new
Date(row.start).toDateString()}</TableCell>
    <TableCell
align="right">{row.marksCount}</TableCell>
    </TableRow>
    <TableCell style={{ paddingBottom: 0, paddingTop:
0 }} colSpan={6}>
        <Collapse in={open[`${index}`]}
timeout="auto" unmountOnExit>
            <Box margin={1}>
                <Typography variant="h6" gutterBottom>
                    Time Marks
                </Typography>
                <Table size="small" aria-
label="purchases">
                    <TableHead>
                        <TableRow>
                            <TableCell>Student
Name</TableCell>
                            <TableCell>Rate</TableCell>
                            <TableCell>Feedback</TableCell>
                        </TableRow>
                        </TableHead>
                        <TableBody>
                            {row.marks.map((item,
index)=>{
                                return(
                                    <TableRow
                                        <TableCell
                                            {item.name}
                                        </TableCell>
                                        <TableCell>{item.rate}</TableCell>
                                        <TableCell>{item.feedback}</TableCell>
                                    </TableRow>
                                )
                            })}
                        </TableBody>
                    </Table>
                </Box>
            </Collapse>
        </TableCell>
    </React.Fragment>
    </TableBody>
    </Table>
}

function CategoryTableBody({rows, isSelected, order, orderBy}) {
    const [open, setOpen] = React.useState({});

```

```

return(
  <TableBody>
    {stableSort(rows, getComparator(order, orderBy))
      .map((row, index) => {
        const isSelected = isSelected(row.name);
        const labelId = `enhanced-table-checkbox-${index}`;

        return (
          <React.Fragment key={index}>
            <TableRow
              hover
              aria-checked={isSelected}
              tabIndex=-1
              key={row.name}
              selected={isSelected}
            >
              <TableCell component="th" scope="row">
                <IconButton aria-label="expand row"
                  size="small" disabled={row.eventsCount === 0} onClick={() => {
                    let newOpen = {...open};
                    newOpen[`_${index}`] =
                      !newOpen[`_${index}`];

                    setOpen({...newOpen})
                  }}>
                  {open[`_${index}`] ?
                    <KeyboardArrowUpIcon /> : <KeyboardArrowDownIcon />}
                </IconButton>
              </TableCell>
              <TableCell component="th" id={labelId}
                scope="row" padding="none">
                {row.name}
              </TableCell>
              <TableCell
                align="right">{row.rate}</TableCell>
              <TableCell
                align="right">{row.eventsCount}</TableCell>
            </TableRow>
            <TableCell style={{ paddingBottom: 0, paddingTop:
              0 }} colSpan={6}>
              <Collapse in={open[`_${index}`]}
                timeout="auto" unmountOnExit>
                <Box margin={1}>
                  <Typography variant="h6" gutterBottom
                    component="div">
                      Events
                    </Typography>
                    <Table size="small" aria-
                      label="purchases">
                      <TableHead>
                        <TableRow>
                          <TableCell>Event
                            Name</TableCell>
                          <TableCell>Rate</TableCell>
                          <TableCell>Start
                            Date</TableCell>
                        </TableRow>
                      </TableHead>
                      <TableBody>
                        {row.categoryEvents.map((item, index)=>{
                          return(
                            <TableRow
                              key={index}>

```

```

component="th" scope="row">
</TableCell>
{item.name}
</TableCell>
<TableCell>{item.rate}</TableCell>
<TableCell>{new
Date(item.start).toDateString()}</TableCell>
</TableRow>
)
)}}
</TableBody>
</Table>
</Box>
</Collapse>
</TableCell>
</React.Fragment>
);
}}
</TableBody>
)
}

```

```

function StudentTableBody({rows, isSelected, order, orderBy}) {
  const [open, setOpen] = React.useState({});
  return (
    <TableBody>
      {stableSort(rows, getComparator(order, orderBy))
        .map((row, index) => {
          const isSelected = isSelected(row.name);
          const labelId = `enhanced-table-checkbox-${index}`;

          return (
            <React.Fragment key={index}>
              <TableRow
                hover
                aria-checked={isSelected}
                tabIndex=-1
                key={row.name}
                selected={isSelected}
              >
                <TableCell component="th" scope="row">
                  <IconButton aria-label="expand row"
                    size="small" disabled={row.marksCount === 0} onClick={() => {
                      let newOpen = {...open};
                      newOpen[`_${index}`] =
!newOpen[`_${index}`];

                      setOpen({...newOpen})
                    }}>
                    {open[`_${index}`] ?
<KeyboardArrowUpIcon /> : <KeyboardArrowDownIcon />}
                  </IconButton>
                </TableCell>
                <TableCell component="th" id={labelId}
                  scope="row" padding="none">
                  {row.name}
                </TableCell>
                <TableCell
                  align="right">{row.rate}</TableCell>
                <TableCell
                  align="right">{row.time}</TableCell>

```

```

        <TableCell
align="right">{row.marksCount}</TableCell>
    </TableRow>
    <TableCell style={{ paddingBottom: 0, paddingTop:
0 }} colSpan={6}>
        <Collapse in={open[`${index}`]}
timeout="auto" unmountOnExit>
            <Box margin={1}>
                <Typography variant="h6" gutterBottom>
                    Events
                </Typography>
                <Table size="small" aria-
label="purchases">
                    <TableHead>
                        <TableRow>
                            <TableCell>Event
Name</TableCell>
                            <TableCell>Rate</TableCell>
                            <TableCell>Time</TableCell>
                            <TableCell>Feedback</TableCell>
                            <TableCell>Status</TableCell>
                        </TableRow>
                    </TableHead>
                    <TableBody>
                        {row.marks.map((item,
index)=>{
                            return(
                                <TableRow
                                key={index}>
                                    <TableCell
component="th" scope="row">
                                        {item.eventName}
                                    </TableCell>
                                    <TableCell>{item.rate}</TableCell>
                                    <TableCell>{item.time}</TableCell>
                                    <TableCell>{item.feedback}</TableCell>
                                    <TableCell>{item.status}</TableCell>
                                </TableRow>
                                )
                            })}
                        </TableBody>
                    </Table>
                </Box>
            </Collapse>
        </TableCell>
    </React.Fragment>
    );
    </TableBody>
)
}

export default function EnhancedTable({type, rows}) {

```

```

const classes = useStyles();
const [order, setOrder] = React.useState('asc');
const [orderBy, setOrderBy] = React.useState('calories');
const [selected, setSelected] = React.useState([]);

const handleRequestSort = (event, property) => {
  const isAsc = orderBy === property && order === 'asc';
  setOrder(isAsc ? 'desc' : 'asc');
  setOrderBy(property);
};

const handleSelectAllClick = (event) => {
  if (event.target.checked) {
    const newSelecteds = rows.map((n) => n.name);
    setSelected(newSelecteds);
    return;
  }
  setSelected([]);
};

const isSelected = (name) => selected.indexOf(name) !== -1;

let body = null;
switch (type) {
  case 'events':
    body = (<EventTableBody rows={rows} isSelected={isSelected}
order={order} orderBy={orderBy}/>);
    break;
  case 'categories':
    body = (<CategoryTableBody rows={rows} isSelected={isSelected}
order={order} orderBy={orderBy}/>);
    break;
  case 'students':
    body = (<StudentTableBody rows={rows} isSelected={isSelected}
order={order} orderBy={orderBy}/>);
    break;
}

return (
  <div className={classes.root}>
    <Paper className={classes.paper}>
      <TableContainer>
        <Table
          className={classes.table}
          aria-labelledby="tableTitle"
          size={'medium'}
          aria-label="enhanced table"
        >
          <EnhancedTableHead
            tableType={type}
            classes={classes}
            numSelected={selected.length}
            order={order}
            orderBy={orderBy}
            onSelectAllClick={handleSelectAllClick}
            onRequestSort={handleRequestSort}
            rowCount={rows.length}
          />
          {body}
        </Table>
      </TableContainer>
    </Paper>
  </div>

```

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2019 р.

ПОСТАЧАЛЬНИК КОНТЕНТУ ДЛЯ НАВЧАЛЬНИХ ПЛАТФОРМ

LEARNSMART TA SMARTBOOK

Програма та методика тестування

ДП.045440-04-51

«ПОГОДЖЕНО»

Керівник проєкту:

_____ Юрій БУХТІЯРОВ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Дмитро БЛАГОДАТНИЙ

ЗМІСТ

1. Об'єкт випробувань.....	3
2. Мета тестування.....	3
3. Методи тестування.....	3
4. Засоби та порядок тестування.....	4

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Вебзастосунок для постачання навчального контенту, що є вебсайтом, серверна частина якого розроблена з використанням фреймворку .NET на основі мови програмування C#, а клієнтська – з використанням фреймворку Microsoft Silverlight.

2. МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- 1) відповідність функціональних можливостей, реалізованих у програмному застосунку до заявлених;
- 2) коректність поведінки програмного застосунку;
- 3) забезпечення належного рівня безпеки даних;
- 4) зручність роботи з вебзастосунком.

3. МЕТОДИ ТЕСТУВАННЯ

Тестування проводитиметься на рівні «системного тестування», тобто буде перевірено кожен модуль розробленого програмного продукту та зв'язки між ними. Для проведення тестування було обрано метод White Box Testing. При такому методі перевіряється як поведінка програмного продукту, так і код.

Використовуються наступні методи:

- 1) Для проведення функціонального тестування Critical path test (тестування критичного шляху);
- 2) Для тестування продуктивності програмного забезпечення load testing (навантажувальне тестування) та stress testing (стресс-тестування);
- 3) тестування інтерфейсу.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Коректність роботи вебзастосунку тестується з використанням таких методів та утиліт:

- 1) введенням недопустимих значень в поля, що можна редагувати;
- 2) ручного тестування відповідності реалізованих функціональних вимог до заявлених;
- 3) тестування при максимальному навантаженні за допомогою утиліти Apache JMeter;
- 4) тестування стабільності роботи при різних умовах за допомогою утиліти Apache JMeter;
- 5) тестування вебзастосунок в різних браузерах;
- 6) тестування зручності користувацького інтерфейсу опитуванням потенційних користувачів.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

_____ Іван ДИЧКА

«___» _____ 2020 р.

ПОСТАЧАЛЬНИК КОНТЕНТУ ДЛЯ НАВЧАЛЬНИХ ПЛАФТОРМ
LEARNSMART ТА SMARTBOOK

Керівництво користувача

ДП.045440-05-34

«ПОГОДЖЕНО»

Керівник проєкту:

_____ Юрій БУХТІЯРОВ

Нормоконтроль:

_____ Микола ОНАЙ

Виконавець:

_____ Дмитро БЛАГОДАТНИЙ

ЗМІСТ

1. Опис структури вебзастосунку	3
2. Процедура авторизації користувача.....	3
3. Процедура конвертування PDF-файлу.....	4
4. Сторінка «Main» та редагування даних про заголовок	5
5. Маркування навчального ресурсу	6
6. Застосування функції Heat Map	7
7. Тестування контенту LO	8
8. Розміщення інтелектуального контенту на навчальних платформах.....	9

1. Опис структури вебзастосунку

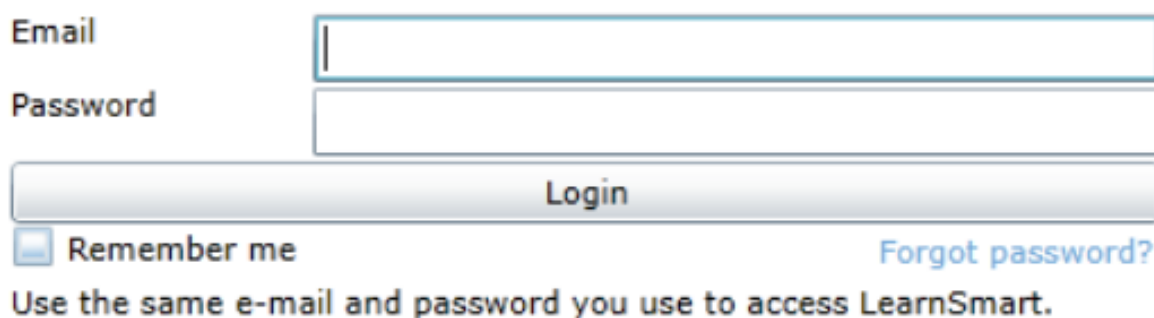
Вебзастосунок для постачання навчального контенту складається з наступних сторінок:

- «Login»;
- «Main»;
- «Convert»
- «Title»;
- «Tagging»;

Перехід між сторінками відбувається за натисканням відповідних кнопок на навігаційній панелі.

2. Процедура авторизації користувача

Сторінка «Login» відкривається за змовчуванням для неавторизованих користувачів (рис. 1). Сторінка містить форму для вводу необхідних для авторизації даних: юзернейма та пароля. Після введення коректних даних та натиснення кнопки «LOGIN», користувач увійде в систему.



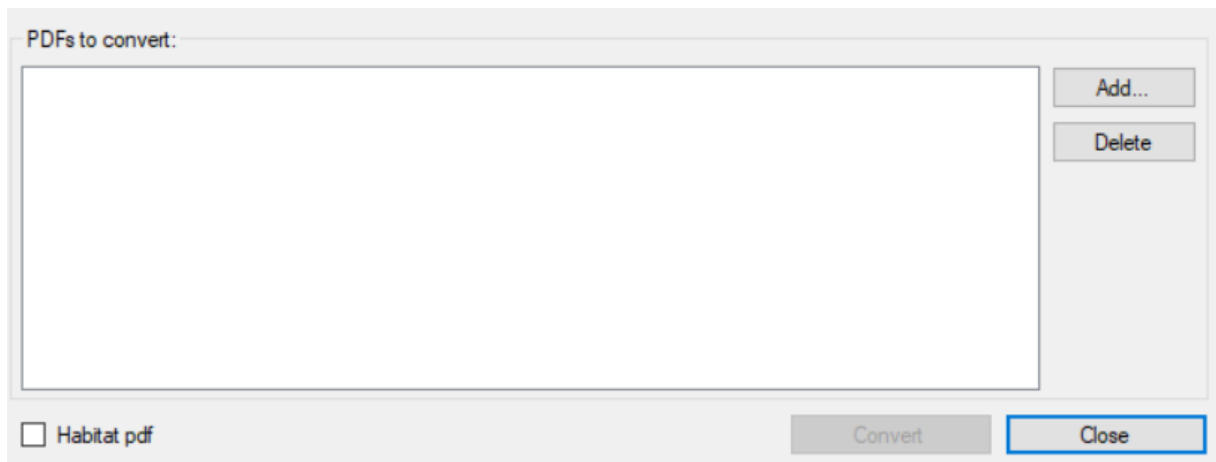
The image shows a login form with the following elements:

- Email** label next to a text input field.
- Password** label next to a text input field.
- A **Login** button below the password field.
- A checkbox labeled **Remember me**.
- A link labeled **Forgot password?** in blue text.
- A footer text: **Use the same e-mail and password you use to access LearnSmart.**

Рис. 1. Сторінка «Login»

3. Процедура конвертування PDF-файлу

Сторінка «Convert» (рис. 2) містить форму для вводу необхідних для конвертації даних: шлях до PDF-файлу, булеве значення, чи перебуває PDF-файл на стадії компіляції. Після введення коректних даних та натиснення кнопки «CONVERT», користувач увійде в систему.

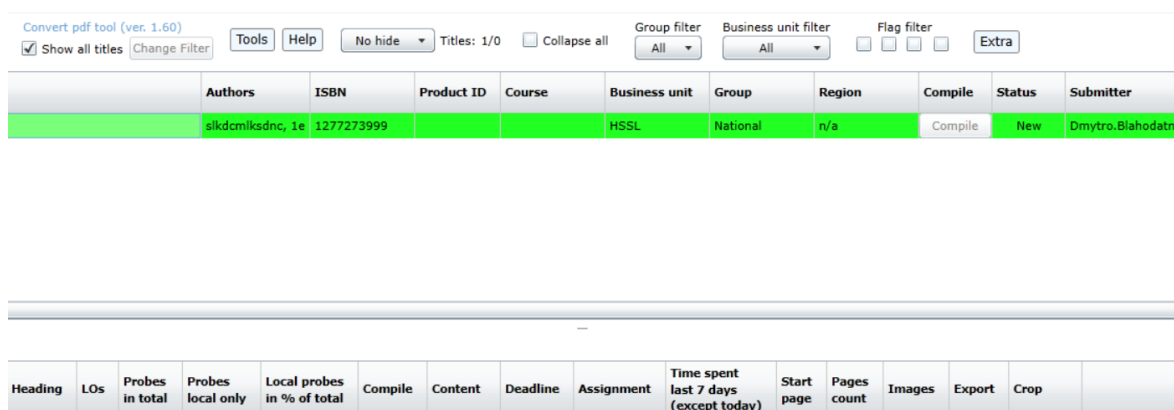


The screenshot shows a window titled "PDFs to convert:". Inside the window is a large empty rectangular box for listing files. To the right of this box are two buttons: "Add..." and "Delete". At the bottom left of the window, there is a checkbox labeled "Habitat pdf". At the bottom right, there are two buttons: "Convert" and "Close". The "Close" button is highlighted with a blue border.

Рис. 2. Сторінка «Convert»

4. Сторінка «Main» та редагування даних про заголовок

Після успішної конвертації, користувач потрапляє на сторінку «Main» (рис. 3). На цій сторінці вже наявні навчальні ресурси та інформація про них. При натисканні на кнопку «TOOLS», відкривається модальне вікно (рис. 4) в якому користувач може редагувати дані про заголовок.

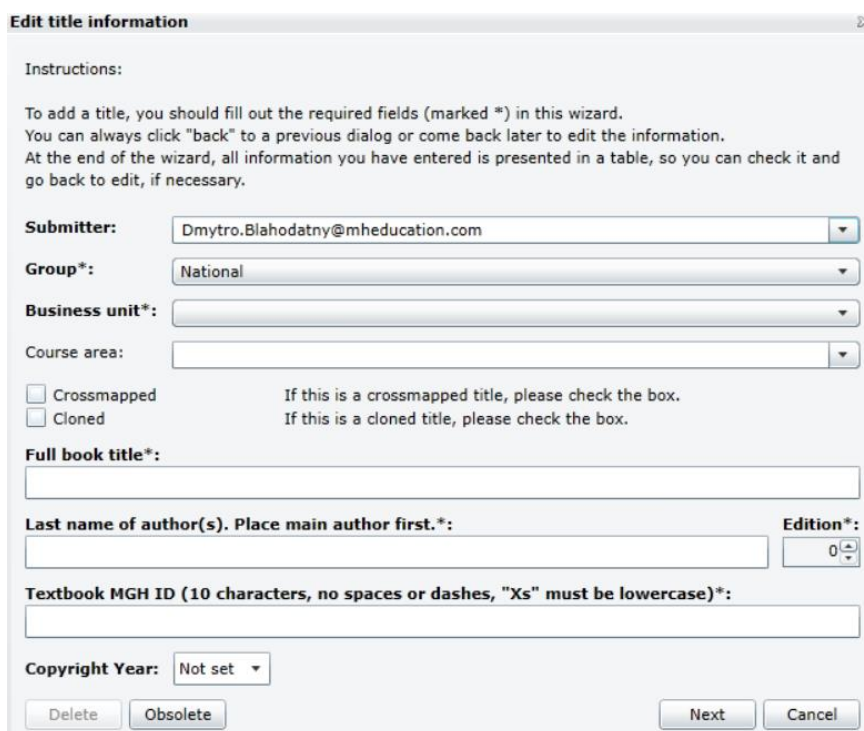


The screenshot shows the 'Main' page of the application. At the top, there is a toolbar with buttons for 'Tools', 'Help', and 'No hide'. Below this is a table with columns: Authors, ISBN, Product ID, Course, Business unit, Group, Region, Compile, Status, and Submitter. The table contains one row of data. Below the table, there is a horizontal separator and another table with columns: Heading, LOs, Probes in total, Probes local only, Local probes in % of total, Compile, Content, Deadline, Assignment, Time spent last 7 days (except today), Start page, Pages count, Images, Export, and Crop.

Authors	ISBN	Product ID	Course	Business unit	Group	Region	Compile	Status	Submitter
sikdcmlksdnc, 1e	1277273999			HSSL	National	n/a	Compile	New	Dmytro.Blahodatny

Heading	LOs	Probes in total	Probes local only	Local probes in % of total	Compile	Content	Deadline	Assignment	Time spent last 7 days (except today)	Start page	Pages count	Images	Export	Crop
---------	-----	-----------------	-------------------	----------------------------	---------	---------	----------	------------	---------------------------------------	------------	-------------	--------	--------	------

Рис. 3. Сторінка «Main»



The screenshot shows the 'Edit title information' modal window. It contains instructions and several input fields for editing title information. The fields include: Submitter (Dmytro.Blahodatny@mheduication.com), Group* (National), Business unit* (empty), Course area (empty), Crossmapped (checkbox), Cloned (checkbox), Full book title* (empty), Last name of author(s). Place main author first.* (empty), Edition* (0), Textbook MGH ID (10 characters, no spaces or dashes, "Xs" must be lowercase)* (empty), and Copyright Year (Not set). There are buttons for 'Delete', 'Obsolete', 'Next', and 'Cancel'.

Edit title information

Instructions:

To add a title, you should fill out the required fields (marked *) in this wizard. You can always click "back" to a previous dialog or come back later to edit the information. At the end of the wizard, all information you have entered is presented in a table, so you can check it and go back to edit, if necessary.

Submitter: Dmytro.Blahodatny@mheduication.com

Group*: National

Business unit*:

Course area:

☐ Crossmapped If this is a crossmapped title, please check the box.
☐ Cloned If this is a cloned title, please check the box.

Full book title*:

Last name of author(s). Place main author first.*: **Edition*:** 0

Textbook MGH ID (10 characters, no spaces or dashes, "Xs" must be lowercase)*:

Copyright Year: Not set

Delete Obsolete Next Cancel

Рис. 4. Модальне вікно для зміни інформації про заголовок

5. Маркування навчального ресурсу

Авторизований користувач може тагувати свої навчальні матеріали (рис. 5). На цій сторінці містяться маркери, за допомогою яких користувач може зазначити частини тексту, до яких будуть присвоєні проби. Після завершення тагування, відкриється модульне вікно (рис. 6), де користувач може власне присвоїти відповідні проби.

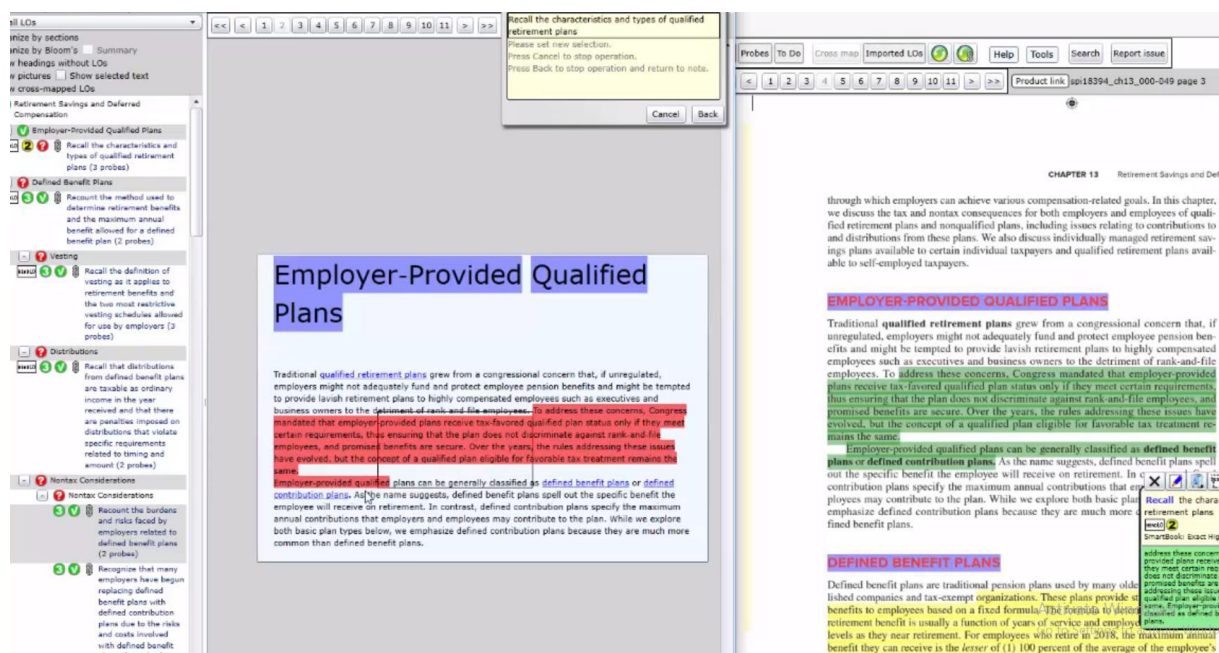


Рис. 5. Маркування навчального ресурсу

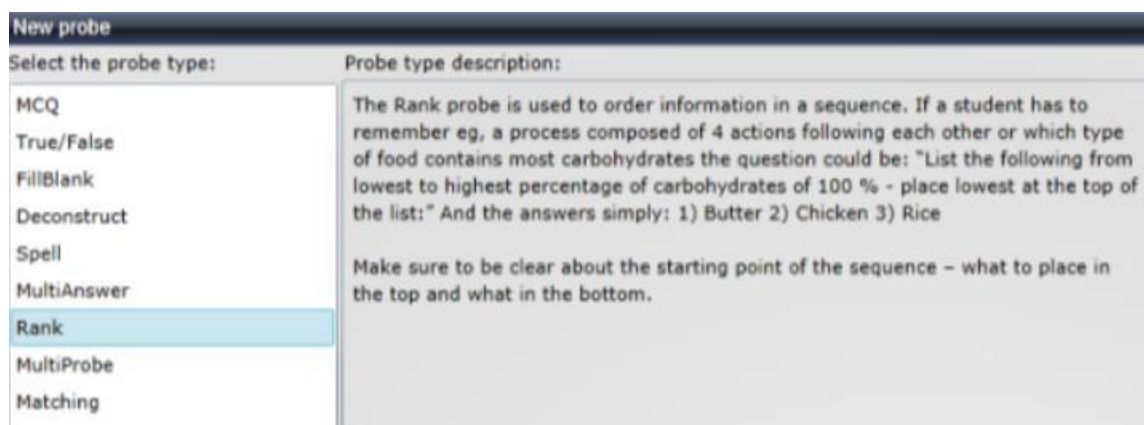


Рис. 6. Вибір проб

6. Застосування функції Heat Map

Успішно присвоївши проби відповідним тегам навчального ресурсу, користувач отримує змогу перевірити успішність учнів на цих самих пробах. Натиснувши кнопку «SEE LOS», та вмикнувши функцію «TURN HEAT MAP ON», користувач може споглядати свого роду звіт про деталізацію тої чи іншої скомпільованої проби. Це можна детально розгледіти за допомогою кольорової гамми (рис. 7).



This will highlight sections of the text according to difficulty, ranging from red (poor student performance aka more difficult) to green (best student performance; less difficult) with a spectrum of shades in between.

Phases of Cellular Respiration

Cellular respiration involves four phases: glycolysis, the preparatory reaction, the citric acid cycle, and the electron transport chain (Fig. 4.2). Glycolysis takes place outside the mitochondria and does not require the presence of oxygen. Therefore, glycolysis is **anaerobic**. The other phases of cellular respiration take place inside the mitochondria, where oxygen is the final acceptor of electrons. Because they require oxygen, these phases are called **aerobic**.

During these phases, notice where CO_2 and H_2O , the end products of cellular respiration, and ATP, the main outcome of respiration, are produced.

Glycolysis (GK: glycol, sugar, andysis, splitting) is the breakdown of glucose (a 6-carbon molecule) to two molecules of pyruvate (two 3-carbon molecules). Oxidation results in NADH and provides enough energy for the net gain of two ATP molecules.

The **preparatory (prep) reaction** takes place in the matrix of the mitochondria. Pyruvate is broken down from a 3-carbon (C_3) to a 2-carbon (C_2) acetyl group, and a 1-carbon CO_2 molecule is released. After glycolysis ends with two molecules of pyruvate, the prep reaction occurs twice per glucose molecule.

with a 4-carbon molecule, forming two 3-carbon citrate molecules. As citrate bonds are broken and oxidation occurs, NADH and FADH₂ are formed, and two CO_2 per citrate are released. The citric acid cycle is able to produce one ATP per turn. Because two acetyl groups enter the cycle per glucose molecule, the cycle turns twice.

The electron transport chain (ETC) is a series of carriers in the matrix of the mitochondria. NADH and FADH₂ give up their high-energy electrons to the chain. Energy is released and captured as the electrons move from a higher-energy to a lower-energy state during each redox reaction. Later, this energy is used for the production of ATP by chemiosmosis. After oxygen receives electrons at the end of the chain, it combines with hydrogen ions (H^+) and becomes water (H_2O).

Pyruvate, the end product of glycolysis, is a pivotal metabolite; its further treatment depends on whether oxygen is available. If oxygen is available, pyruvate enters a mitochondrion and is broken down completely to CO_2 and H_2O as shown in the cellular respiration equation (p. 134). If oxygen is not available, pyruvate is further metabolized in the cytoplasm by an anaerobic process called **fermentation**. Fermentation results in a net gain of only two ATP per glucose molecule.

Check Your Progress

8.1

1. Explain the benefit of slow glucose breakdown rather than rapid breakdown during cellular respiration.
2. Describe the four phases of complete glucose breakdown, including which release CO_2 and which produce H_2O .

Red – 60% or fewer students answered it correctly
Orange – between 60-70% of students answered it correctly
Yellow – between 70-80% of students answered it correctly
Green – More than 80% of students answered it correctly

Рис. 7. Візуалізація Heat Map функції

7. Тестування контенту LO

Після успішної компіляції проб, користувач отримує доступ до додаткової функціональності системи. Однією з основних функціональностей є тестування навчального ресурсу. Її можна застосувати, натиснувши кнопку «TEST CONTENT» (рис. 8). За допомогою цієї функції користувач може перевірити на валідність кожну з окремих проб, або весь розділ цілком. В кінці тестування буде згенеровано відповідний звіт (рис. 9).

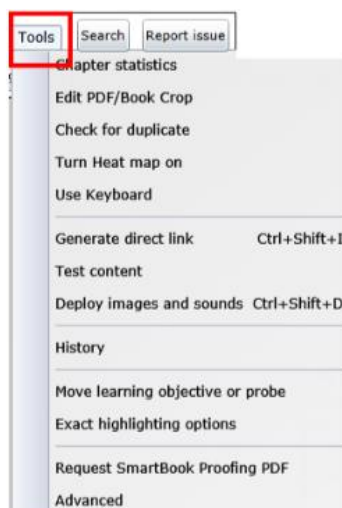
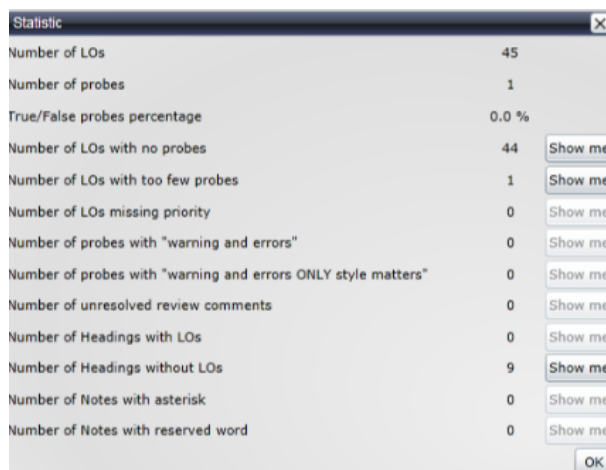


Рис. 8. Тестування контенту



Number of LOs	45	
Number of probes	1	
True/False probes percentage	0.0 %	
Number of LOs with no probes	44	Show me
Number of LOs with too few probes	1	Show me
Number of LOs missing priority	0	Show me
Number of probes with "warning and errors"	0	Show me
Number of probes with "warning and errors ONLY style matters"	0	Show me
Number of unresolved review comments	0	Show me
Number of Headings with LOs	0	Show me
Number of Headings without LOs	9	Show me
Number of Notes with asterisk	0	Show me
Number of Notes with reserved word	0	Show me

Рис. 9. Звіт системи щодо валідності проб

8. Створення, редагування та видалення карток-задач

На оновленій головній сторінці «Main» знаходяться записи з успішно скомпільованими навчальними ресурсами. Користувач отримує змогу провести деталізацію кожного із окремих маркувань, та налаштувати пріоритет розміщення того чи іншого навчального ресурсу (рис. 10). Після валідного розміщення інтелектуального ресурсу на навчальній платформі, інструктор має змогу поділитися ним зі своїми студентами, та почати будувати унікальний навчальний процес (рис. 11).

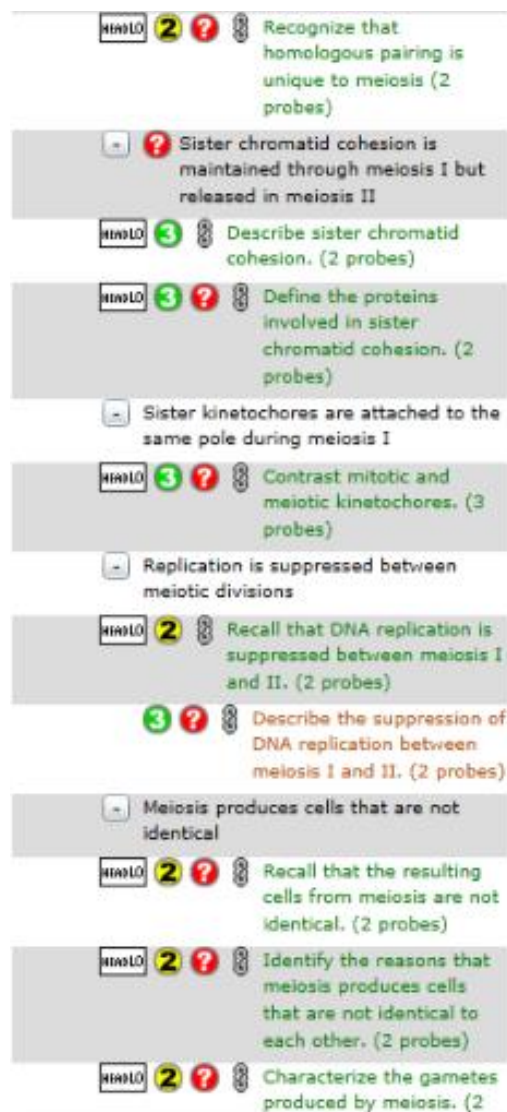


Рис. 10. Статус маркувань під час процесу розміщення на навчальних платформах

Theatre - The Creative Spirit: An Introduction to Theatre - Arnold, 6e	
7	Understanding Style: Realism
8	Expressing a Worldview through Realism
9	Understanding Style: Theatricalism
10	Expressing a Worldview through Theatricalism
11	The Musical Theatre
12	The Elements of Drama and Dramatic Structure
13	Innovations in Dramatic Structure
14	Genre
Read Only	
TOC	Table of Contents and Preface
15	The Project
APP	Appendix: Guided Writing Assignments
NOT	Notes
GLO	Glossary
CRE	Credits
IND	Index

This is chapter 15 of the book but there are no probes for it, so it is set up as a Supplement > Other module and sits in the Read-only content pack.

Рис. 11. Розміщення навчальних ресурсів на платформах